Test Plan for Liberty Alliance SAML Test Event Test Criteria SAML 2.0

Version 3.2.2

Editor:

Kyle Meadors, Drummond Group Inc.

Abstract:

This document describes the test steps to achieve the Liberty Interoperable™ designation for various SAML 2.0 modes and profiles.

Filename:

Liberty_Interoperability_SAML_Test_Plan_v3.2.2.odt

Contents

Introduction

Overview of Test Plan

This document is the Liberty SAML 2.0 Test Criteria Test Plan, which contains the scope of the technical requirements for Liberty certification of SAML 2.0. This document is intended to be publicly viewable through the Liberty Alliance website as well as prospective test participants. The document is reviewed and authored by the Technology Expert Group (TEG)

The contents of this document include the test cases for Liberty SAML 2.0 certification as well as additional technical information relevant to testing. The test cases include different test steps, which as a whole cover the requirements of the SAML profiles [SAMLProf] and SAML conformance modes [SAMLConf].

Another document, Liberty SAML 2.0 Process Test Plan, contains the detailed testing process and test administration requirements for the SAML 2.0 certification test. The Liberty SAML 2.0 Process Test Plan is available only to registered test participants. While the Process Test Plan is used in completing a certification event, it is not needed to understand the technical expectation for completing SAML 2.0 certification.

Test Plan History

This test plan replaces SAML 2.0 Interoperability Testing Procedure (vs. 3.1) test plan [SAMLTP31]. The major changes to this version are modifications to the eGov profile and removing the ECP Conformance mode testing requirements. Also, consensus items reached from the last interoperability test event have been included here.

SAML 2.0 Interoperability Testing Procedure, vs. 3.1 (07/15/2008)

SAML 2.0 Interoperability Testing Procedure, vs. 3.0.J (11/20/2007)

SAML 2.0 Interoperability Testing Procedure, vs. 2.0 (07/07/2006)

SAML 2.0 Interoperability Testing Procedure, vs. 1.0 (2005)

SAML Conformance Modes

This test plan document contains test cases that cover the many of the operational conformance modes of SAML 2.0 and the specific features that are required or optional for each mode. The details of each mode are provided in [SAMLConf], and the conformance modes a listed here:

IdP – Identity Provider

IdP Lite – Identity Provider Lite

SP – Service Provider

SP Lite – Service Provider Lite

IdP Extended – Identify Provider Extended

SP Extended – Service Provider Extended

SAML Attribute Authority (Requester/Responder)

SAML Authorization Decision Authority (Requester/Responder)

SAML Authentication Authority (Requester/Responder)

Each conformance mode requires different test cases, but some test cases cover multiple conformance modes. The required test cases for each conformance mode are noted in the Test Case section of this document.

Certification in conformance modes IdP Extended and SP Extended can only be given if a participant has met the certification requirements of IdP mod and SP mode, respectively.

eGov 1.5 Profile

The eGov 1.5 Profile is a conformance profile developed by Liberty eGovernment SIG . The test cases within this test plan to achieve eGov certification are based on the requirements stated in the eGov 1.5 profile. The eGov 1.5 profile and other associated documents should be consulted for further explanation of the eGov requirements.

http://www.projectliberty.org/liberty/strategic_initiatives/egovernment

POST Binding

Although the POST binding is not included in the SAML SCR, it is permitted with the SAML specification and has some user deployment. POST Binding is an optional Liberty designation conformance mode. It involves use of POST binding for AuthnRequest, Name ID Management and SLO. Certification in the POST Binding mode is done through successfully completing this Test Case E – POST Binding.

Technical Requirements

Metadata

There are no normative requirements in [SAMLConf] regarding the content or processing of metadata as described in [SAMLMeta]. However, for purposes of this certification event, implementations are required to:

Furnish correct metadata, and

Process metadata furnished by other testing partners

While metadata is not specified for SAML Attribute Requesters, interoperability with SAML Authorities is very difficult without it, and for this certification event it is required that SAML Attribute Requesters provide metadata as described in the draft metadata extension specification [SAMLMetaExt].

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, it is required that IdP implementations offer at least one of these authentication methods:

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

Similarly, it is required that user agents be able to authenticate using at least one of these methods.

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.

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 comparison decisions up to the implementation [SAMLCore]. However, for purposes 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. The overall concept of the testing of the Authentication Authority is to create several different assertions using different authentication contexts. Then these are queried using the query terms (“exact”, “better”, “maximum”, “minimum”) and a reference authentication context.

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 authentication context processing rules.

Name Identifier Formats

The following Name Identifier 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 last two be processed.

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. Single Logout: The <LogoutRequest> and <LogoutResponse> MUST be signed for the HTTP redirect binding.
3. Name Identifer Management: The <ManageNameIDRequest> and <ManageNameIDResponse> MUST be signed for the HTTP redirect binding.

Note that when a test step refers to a “signed SAML Response message” this implies the assertion element itself is signed per the requirements in [SAMLProf].

SP and IdP implementations may indicate via metadata a desire for requests or responses to be signed for other bindings than those indicated above. While such stipulations in metadata may not be binding, participants are strongly encouraged to adhere to these requests and may be required to do so to insure interoperability.

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, these elements MUST 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]. For this certification event, this most common practice stated above SHOULD be done.

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.

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] except for X.500/LDAP, which is described in [SAMLLDAP]:

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

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

Consensus Items

Consensus Items contains standards/implementation issues the product test group reached consensus on in previous Liberty test events in order to achieve interoperability among those product test groups. In order to maintain interoperability with previously tested versions, the consensus items will be observed in this test event.

In an authentication request message, an interoperable implementation must accept a requested authentication context listed in the <RequestedAuthnContext> element if it can

meet the authentication context requirements of the specified element and not require that such information be specified out-of-band.

DSAwithSHA1 signature algorithm not supported. Section 4.1 of [SAMLConf] states that the DSAwithSHA1 signature algorithm, while recommended, is not required by SAML 2.0. Participants are only to use digital certificates with the required RSAwithSHA1 signature algorithm.

Ignore EncryptionMethod elements in metadata. There is some confusion of interpretation implementation of the EncryptionMethod metadata elements described in Section 2.4.1.1 of [SAMLMeta]. After confirming with OASIS SSTC, EncryptionMethod is to be ignored.

Encryption with NameIDPolicy and ID Encryption. A question had arisen on interpreting NameIDPolicy from [SAMLCore] in lines 2136-2142. It was decided that if NameIDPolicy of AuthnRequest says ID is to be encrypted, it must be encrypted in the assertion and if NameIDPolicy of AuthnRequest does not state the ID is to be encrypted, the IDP MAY still encrypt the ID based on its policy, specifically its policy with the SP.

SSL Server-side Authentication Only for SOAP connections. To insure all participants used the same security settings, it was agreed to only use SSL server-side authentication for SOAP connections and not to use SSL client-side authentication.

Test Cases

Overview of Test Case Description

Each test case is setup with the first part listing an overview of the test steps in the test case. The second part describes the details of the individual test steps to carry out the test case. The test step overview lists the sequence of test steps along with a general description of the message or action or configuration setting required. The test step details provide more information on the expected test steps.

Test Cases Associated with Conformance Modes

In order to achieve certification in one or more of the Liberty SAML Conformance Modes, the associated test cases must be completed with all test participants with aligning modes. For example, a product testing for an IdP conformance mode must complete Test Cases A, B, C, D, H, I, J, K, L and P against all products testing for a SP conformance mode and SP Lite conformance mode. The specific pairing among participants will be given at the beginning of the certification event. A conformance mode may not require completion of all the test steps in the associated test cases. The individual test cases provide details of test steps that may or must be omitted depending on the conformance mode.

Conformance Mode	Test Cases
IdP	A, B, C, D, H, I, J, K, L, P
IdP Extended	F, G
IdP Lite	A, B, H, I, J, K, L, P
SP	A, B, C, D, H, I, J, K, L, P
SP Extended	F, G
SP Lite	A, B, H, I, J, K, L, P
POST	E, P
SAML Attribute Authority (Requester/Responder)	N
SAML Authorization Decision Authority (Requester/Responder)	O
SAML Authentication Authority (Requester/Responder)	M
eGov 1.5 profile	A, B, H, I, J, K, L, P, Q, R, S, T

Test Case A: Web SSO and SLO – Redirect Binding

Preconditions:

Metadata exchanged and loaded
Encryption disabled
User Identities Not Federated

Conformance Modes: IdP, SP, IdP Lite, SP Lite, eGov

Step 1: AuthnRequest, Redirect Binding, Federate
Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
IdP CONFIRM: Name ID format is 'persistent'.

Step 2: Assertion Response, POST binding
Description: User provides assigned credentials for authentication. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.
SP CONFIRM: Valid assertion is returned from IdP.
SP CONFIRM: User identity has been federated with IdP.
IdP CONFIRM: User identity has been federated with SP.

Step 3: SLO Request, IdP-Initiated, Redirect Binding
Description: IdP logs out User session. IdP sends a signed LogoutRequest message to SP using HTTP Redirect binding.SP logs out User session. SP returns a signed LogoutResponse message to IdP using HTTP Redirect binding.

SP CONFIRM: Receives signed LogoutRequest through HTTP Redirect binding.
SP CONFIRM: User logged out at SP.
IdP CONFIRM: Receives signed LogoutResponse through HTTP Redirect binding.
IdP CONFIRM: User logged out at IdP.

Step 4: AuthnRequest, Redirect Binding, Already Federated

Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with AllowCreate is set to FALSE. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.

IdP CONFIRM: Name ID format is 'persistent'.

Step 5: Assertion Response, POST binding

Description: User provides assigned credentials for authentication. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

SP CONFIRM: Valid assertion is returned from IdP.

SP CONFIRM: User identity has been federated with IdP.

IdP CONFIRM: User identity has been federated with SP.

Step 6: SLO Request, SP-Initiated, Redirect Binding

Description: SP logs out User session. SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP logs out User session. IdP returns a signed LogoutResponse message to SP using HTTP Redirect binding.

SP CONFIRM: User logged out at SP.

IdP CONFIRM: Receives signed LogoutRequest through HTTP Redirect binding.

IdP CONFIRM: User logged out at IdP.

SP CONFIRM: Receives signed on LogoutResponse through HTTP Redirect binding.

Test Case B: Web SSO – Artifact Binding and SLO – SOAP Binding

Preconditions:

Metadata exchanged and loaded
Encryption enabled for Assertions
Encryption enabled for NameIDs in SLO messages
User Identities Not Federated

NOTE: The SAML Conformance specification states that SOAP Binding for SLO is optional for SP Lite and IdP Lite applications. SP Lite and IdP Lite participants may choose to use Redirect Binding for test steps preforming SLO actions instead of SOAP Binding.

Conformance Modes: IdP, SP, IdP Lite, SP Lite, eGov

Step 1: AuthnRequest, Redirect Binding, Federate
Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
IdP CONFIRM: Name ID format is 'persistent'.

Step 2: Assertion Response, HTTP Artifact
Description: User provides assigned credentials for authentication. IdP creates assertion of User. <Response> message is associated with an artifact. IdP returns artifact in a through HTTP Redirect binding.

SP CONFIRM: Artifact is sent by IdP.
IdP CONFIRM: User identity has been federated with SP.

Step 3: Artifact Resolution, SOAP Binding
Description: SP sends ArtifactResolve message to IdP referencing artifact through synchronous SOAP binding. IdP confirms artifact and returns <Response> message to SP in ArtifactResponse message.

SP CONFIRM: Receives ArtifactResponse message containing <Response> message with signed assertion of User.
SP CONFIRM: User identity has been federated with IdP.
IdP CONFIRM: Receives ArtifactResolve message.

Step 4: SLO Request, IdP-Initiated, SOAP Binding
Description: IdP logs out User session. IdP sends a signed LogoutRequest message to SP using synchronous SOAP binding. SP logs out User session. SP returns a signed LogoutResponse message to IdP using synchronous SOAP binding.

IdP CONFIRM: User logged out at IdP.
SP CONFIRM: Receives signed LogoutRequest through SOAP binding.
SP CONFIRM: User logged out at SP.

340 IdP CONFIRM: Receives signed LogoutResponse through SOAP binding.

Step 5: Redirect Binding, Already Federated

Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with AllowCreate is set to FALSE. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.

IdP CONFIRM: Name ID format is 'persistent'.

Step 6: Assertion Response, HTTP Artifact

Description: User provides assigned credentials for authentication. IdP creates assertion of User. <Response> message is associated with an artifact. IdP returns artifact in a through HTTP Redirect binding.

SP CONFIRM: Artifact is sent by IdP.

IdP CONFIRM: User identity has been federated with SP.

Step 7: Artifact Resolution, SOAP Binding

Description: SP sends ArtifactResolve message to IdP referencing artifact through synchronous SOAP binding. IdP confirms artifact and returns <Response> message to SP in ArtifactResponse message.

SP CONFIRM: Receives ArtifactResponse message containing <Response> message with signed assertion of User.

SP CONFIRM: User identity has been federated with IdP.

IdP CONFIRM: Receives ArtifactResolve message.

Step 8: SLO Request, SP-Initiated, SOAP Binding

Description: SP logs out User session. SP sends a signed LogoutRequest message to IdP using synchronous SOAP binding. IdP logs out User session. IdP returns a signed LogoutResponse message to SP using synchronous SOAP binding.

SP CONFIRM: User logged out at SP.

IdP CONFIRM: Receives signed LogoutRequest through SOAP binding.

IdP CONFIRM: User logged out at IdP.

SP CONFIRM: Receives signed on LogoutResponse through SOAP binding.

Test Case C – NameID Management – Redirect Binding

Preconditions:

Metadata exchanged and loaded
Encryption disabled
User Identities Not Federated

Conformance Modes: IdP, SP

Step 1: AuthnRequest, Redirect Binding, Federate
Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
IdP CONFIRM: Name ID format is 'persistent'.

Step 2: Assertion Response, POST binding
Description: User provides assigned credentials for authentication. IdP provides assertion of User and IdP returns a SAML Response message through HTTP POST binding.

SP CONFIRM: IdP returns SAML Response through HTTP POST binding.
SP CONFIRM: Receives signed assertion is returned from IdP.
SP CONFIRM: User identity has been federated with IdP.
IdP CONFIRM: User identity has been federated with SP.

Step 3: MNI Request, IdP-Initiated, Redirect binding
Description: IdP sends signed ManageNameIdRequest message requesting to use a new NameID (value chosen by the IdP at time of test execution) for the User to the SP using HTTP Redirect binding. SP accepts the new NameID for the User. SP returns signed ManageNameIdResponse message using HTTP Redirect binding.

SP CONFIRM: Receives signed ManageNameIdRequest on HTTP Redirect binding.
SP CONFIRM: New NameID is accepted.
IdP CONFIRM: Receives signed ManageNameIdResponse on HTTP Redirect binding.

Step 4: SLO Request, SP-Initiated, Redirect Binding
Description: SP logs out User session. SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP logs out User session. IdP returns a signed LogoutResponse message to SP using HTTP Redirect binding.

SP CONFIRM: User logged out at SP.
IdP CONFIRM: Receives signed LogoutRequest through HTTP Redirect binding.
IdP CONFIRM: New NameID from Step 3 is used in LogoutRequest.
IdP CONFIRM: User logged out at IdP.
SP CONFIRM: Receives signed on LogoutResponse through HTTP Redirect binding.

Step 5: AuthnRequest, Redirect Binding, Already Federated

Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with AllowCreate is set to FALSE. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.

IdP CONFIRM: Name ID format is 'persistent'.

Step 6: Assertion Response, POST binding

Description: User provides assigned credentials for authentication. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

SP CONFIRM: Valid assertion is returned from IdP.

SP CONFIRM: User identity has been federated with IdP.

IdP CONFIRM: User identity has been federated with SP.

Step 7: MNI Request, SP-Initiated, Redirect binding

Description: SP sends signed ManageNameIdRequest message requesting to use a new NameID (value chosen by the SP at time of test execution) for the User to the IdP using HTTP Redirect binding. IdP accepts the new NameID for the User. IdP returns signed ManageNameIdResponse message using HTTP Redirect binding.

IdP CONFIRM: Receives signed ManageNameIdRequest on HTTP Redirect binding.

IdP CONFIRM: New NameID is accepted.

SP CONFIRM: Receives signed ManageNameIdResponse on HTTP Redirect binding.

Step 8: SLO Request, IdP-Initiated, Redirect Binding

Description: IdP logs out User session. IdP sends a signed LogoutRequest message to SP using HTTP Redirect binding. SP logs out User session. SP returns a signed LogoutResponse message to IdP using HTTP Redirect binding.

IdP CONFIRM: User logged out at IdP.

SP CONFIRM: Receives signed LogoutRequest through HTTP Redirect binding.

SP CONFIRM: New NameID from Step 7 is used in LogoutRequest.

SP CONFIRM: User logged out at SP.

IdP CONFIRM: Receives signed LogoutResponse through HTTP Redirect binding.

Step 9: AuthnRequest, Redirect Binding, Already Federated

Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with AllowCreate is set to FALSE. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.

IdP CONFIRM: Name ID format is 'persistent'.

Step 10: Assertion Response, POST binding

Description: User provides assigned credentials for authentication. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

SP CONFIRM: Valid assertion is returned from IdP.

SP CONFIRM: User identity has been federated with IdP.

IdP CONFIRM: User identity has been federated with SP.

Step 11: MNI-Terminate from SP

Description: SP sends signed ManageNameIdRequest message with the <Terminate> element to the IdP using HTTP Redirect binding. Federation for User is terminated. IdP returns signed ManageNameIdResponse message using HTTP Redirect binding.

IdP CONFIRM: Receives signed ManageNameIdRequest with <Terminate> element on HTTP Redirect binding.

IdP CONFIRM: Federation of User is terminated.

SP CONFIRM: Receives signed ManageNameIdResponse on HTTP Redirect binding.

SP CONFIRM: Federation of User is terminated.

Test Case D – NameID Management – SOAP Binding

Preconditions:

Metadata exchanged and loaded
Encryption enabled for Assertions
Encryption enabled for NameIDs in MNI messages
Encryption enabled for NameIDs in SLO messages
User Identities Not Federated

NOTE: The SAML Conformance specification states that SOAP Binding for MNI is optional for SP applications. SP participants may choose to use Redirect Binding for test steps preforming MNI actions instead of SOAP Binding.

Conformance Modes: IdP, SP

Step 1: AuthnRequest, Redirect Binding, Federate
Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
IdP CONFIRM: Name ID format is 'persistent'.

Step 2: Assertion Response, HTTP Artifact
Description: User provides assigned credentials for authentication. IdP creates assertion of User. <Response> message is associated with an artifact. IdP returns artifact in a through HTTP Redirect binding. SP sends ArtifactResolve message to IdP referencing artifact through synchronous SOAP binding. IdP confirms artifact and returns <Response> message to SP in ArtifactResponse message.

SP CONFIRM: Artifact is sent by IdP.
IdP CONFIRM: User identity has been federated with SP.

Step 3: Artifact Resolution, SOAP Binding
Description:

SP CONFIRM: Receives ArtifactResponse message containing <Response> message with signed assertion of User.
SP CONFIRM: User identity has been federated with IdP.
IdP CONFIRM: Receives ArtifactResolve message.

Step 4: MNI Request, SP-Initiated, SOAP binding
Description: SP sends signed ManageNameIdRequest message requesting to use a new NameID (value chosen by the SP at time of test execution) for the User to the IdP using SOAP binding. IdP accepts the new NameID for the User. IdP returns signed ManageNameIdResponse message using same synchronous SOAP binding.

IdP CONFIRM: Receives signed ManageNameIdRequest on SOAP binding.
IdP CONFIRM: New NameID is accepted.
SP CONFIRM: Receives signed ManageNameIdResponse on SOAP binding.

Step 5: SLO Request, IdP-Initiated, SOAP Binding
Description: IdP logs out User session. IdP sends a signed LogoutRequest message to SP using synchronous SOAP binding. SP logs out User session. SP returns a signed LogoutResponse message to IdP using synchronous SOAP binding.

IdP CONFIRM: User logged out at IdP.
SP CONFIRM: Receives signed LogoutRequest through SOAP binding.
SP CONFIRM: User logged out at SP.
IdP CONFIRM: Receives signed LogoutResponse through SOAP binding.

Step 6: Redirect Binding, Already Federated
Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with AllowCreate is set to FALSE. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
IdP CONFIRM: Name ID format is 'persistent'.

Step 7: Assertion Response, HTTP Artifact
Description: User provides assigned credentials for authentication. IdP creates assertion of User. <Response> message is associated with an artifact. IdP returns artifact in a through HTTP Redirect binding.

SP CONFIRM: Artifact is sent by IdP.
IdP CONFIRM: User identity has been federated with SP.

Step 8: Artifact Resolution, SOAP Binding
Description: SP sends ArtifactResolve message to IdP referencing artifact through synchronous SOAP binding. IdP confirms artifact and returns <Response> message to SP in ArtifactResponse message.

SP CONFIRM: Receives ArtifactResponse message containing <Response> message with signed assertion of User.
SP CONFIRM: User identity has been federated with IdP.
IdP CONFIRM: Receives ArtifactResolve message.

Step 9: MNI Request, IdP-Initiated, SOAP binding

Description: IdP sends signed ManageNameIdRequest message requesting to use a new NameID (value chosen by the IdP at time of test execution) for the User to the SP using SOAP binding. SP accepts the new NameID for the User. SP returns signed ManageNameIdResponse message using same synchronous SOAP binding.

SP CONFIRM: Receives signed ManageNameIdRequest on HTTP Redirect binding.
SP CONFIRM: New NameID is accepted.
IdP CONFIRM: Receives signed ManageNameIdResponse on HTTP Redirect binding.

Step 10: SLO Request, SP-Initiated, SOAP Binding

Description: SP logs out User session. SP sends a signed LogoutRequest message to IdP using synchronous SOAP binding. IdP logs out User session. IdP returns a signed LogoutResponse message to SP using synchronous SOAP binding.

SP CONFIRM: User logged out at SP.
IdP CONFIRM: Receives signed LogoutRequest through SOAP binding.
IdP CONFIRM: User logged out at IdP.
SP CONFIRM: Receives signed on LogoutResponse through SOAP binding.

Step 11: Redirect Binding, Already Federated

Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with AllowCreate is set to FALSE. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
IdP CONFIRM: Name ID format is 'persistent'.

Step 12: Assertion Response, HTTP Artifact

Description: User provides assigned credentials for authentication. IdP creates assertion of User. <Response> message is associated with an artifact. IdP returns artifact in a through HTTP Redirect binding.

SP CONFIRM: Artifact is sent by IdP.
IdP CONFIRM: User identity has been federated with SP.

Step 13: Artifact Resolution, SOAP Binding

Description: SP sends ArtifactResolve message to IdP referencing artifact through synchronous SOAP binding. IdP confirms artifact and returns <Response> message to SP in ArtifactResponse message.

SP CONFIRM: Receives ArtifactResponse message containing <Response> message with signed assertion of User.

SP CONFIRM: User identity has been federated with IdP.

IdP CONFIRM: Receives ArtifactResolve message.

Step 14: MNI-Terminate, IdP-Initiated

Description: IdP sends signed ManageNameIdRequest message with the <Terminate> element to the IdP using SOAP binding. Federation for User is terminated. IdP returns signed ManageNameIdResponse message using same synchronous binding.

SP CONFIRM: Receives signed ManageNameIdRequest with <Terminate> element on SOAP binding.

SP CONFIRM: Federation of User is terminated.

IdP CONFIRM: Receives signed ManageNameIdResponse on SOAP binding.

IdP CONFIRM: Federation of User is terminated.

Test Case E – POST Binding

Preconditions:

Metadata exchanged and loaded
Encryption disabled
User Identities Not Federated

Conformance Modes: POST Binding

Step 1: SSO, Federate, POST Binding
Description: User does Single Sign-On at SP with Persistent Name Identifier and AllowCreate set to TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP POST binding.
IdP CONFIRM: User has been federated
SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

Step 2: MNI Request, IdP-Initiated, POST binding
Description: IdP sends signed ManageNameIdRequest message to the SP using HTTP POST binding. SP returns signed ManageNameIdResponse message using HTTP POST binding.

SP CONFIRM: Receives signed ManageNameIdRequest on HTTP POST binding.
IdP CONFIRM: Receives signed ManageNameIdResponse on HTTP POST binding.

Step 3: SLO Request, SP-Initiated, POST Binding
Description: SP sends a signed LogoutRequest message to IdP using HTTP POST binding. IdP logs out User session. IdP returns a signed LogoutResponse message.

IdP CONFIRM: Receives signed LogoutRequest on HTTP POST binding.
SP CONFIRM: Receives signed LogoutResponse on HTTP POST binding.

Step 3: SSO, Already Federated, POST Binding
Description: User does Single Sign-On at SP with AllowCreate set to FALSE. SP communication to the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP POST binding.
SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

Step 4: SLO Request, IdP-Initiated, POST Binding
Description: IdP logs out User session. IdP sends a signed LogoutRequest message to SP using HTTP POST binding. SP returns a signed LogoutResponse message.

IdP CONFIRM: Receives signed LogoutRequest on HTTP POST binding.
SP CONFIRM: Receives signed LogoutResponse on HTTP POST binding.

Step 5: SSO, Already Federated, POST Binding
Description: User does Single Sign-On at SP with AllowCreate set to FALSE. SP communication to the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.
IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP POST binding.
SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

Step 6: MNI-Terminate, IdP-Initiated
Description: IdP sends signed ManageNameIdRequest message with the Terminate element to the SP using HTTP POST binding. Federation for User is terminated. SP returns signed ManageNameIdResponse message using HTTP POST binding.
SP CONFIRM: Receives signed ManageNameIdRequest with Terminate flag on HTTP POST binding.
SP CONFIRM: Federation of User is terminated.
IdP CONFIRM: Receives signed ManageNameIdResponse on HTTP POST binding.
IdP CONFIRM: Federation of User is terminated.

Step 7: SSO, Federate, POST Binding
Description: User does Single Sign-On at SP with Persistent Name Identifier and AllowCreate set to TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.
IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP POST binding.
IdP CONFIRM: User has been federated
SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

Step 8: MNI Request, SP-Initiated, POST binding
Description: SP sends signed ManageNameIdRequest message to the IdP using HTTP POST binding. IdP returns signed ManageNameIdResponse message using HTTP POST binding.
IdP CONFIRM: Receives signed ManageNameIdRequest on HTTP POST binding.
SP CONFIRM: Receives signed ManageNameIdResponse on HTTP POST binding.

Step 9: SLO Request, IdP-Initiated, POST Binding
Description: IdP sends a signed LogoutRequest message to SP using HTTP POST binding. SP logs out User session. SP returns a signed LogoutResponse message.
SP CONFIRM: Receives signed LogoutRequest on HTTP POST binding.
IdP CONFIRM: Receives signed LogoutResponse on HTTP POST binding.

Step 10: SSO, Already Federated, POST Binding

Description: User does Single Sign-On at SP with AllowCreate set to FALSE. SP communication to the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP POST binding.

SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

Step 11: SLO Request, SP-Initiated, POST Binding

Description: SP sends a signed LogoutRequest message to IdP using HTTP POST binding. IdP logs out User session. IdP returns a signed LogoutResponse message.

IdP CONFIRM: Receives signed LogoutRequest on HTTP POST binding.

SP CONFIRM: Receives signed LogoutResponse on HTTP POST binding.

Test Case F – IdP Proxy

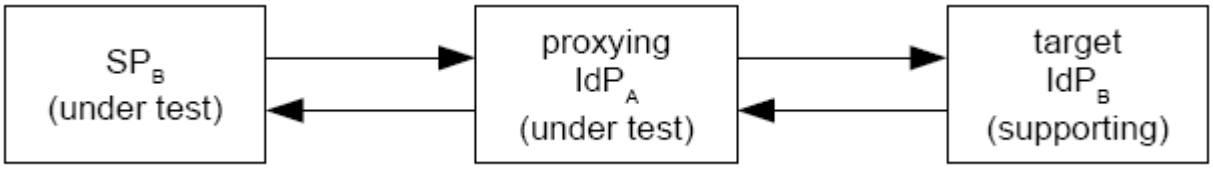
Preconditions:

Metadata exchanged and loaded
Encryption disabled
User Identities Not Federated

Conformance Modes: IdP Extended, SP Extended

Background on IdP Proxy

Refer to Section 3.4.1.5 of [SAMLCore] for more background. The IdP Proxy feature requires two IdP implementations and one SP implementation. If we have participants 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:



To complete this Test Case, the IdP under test must receive an authentication request for a User it can not authenticate but a User that the supporting IdP can authenticate. This coordination of User accounts must be done prior to executing the test case.

Step 1: ProxyCount=0

Description: SP sets ProxyCount=0 where proxy is disallowed.

SP CONFIRM: SP has disallowed proxy.

Step 2: AuthnRequest from SP to IdP_A, Redirect Binding, Federate

Description: User/SP attempts Single Sign-On with Persistent Name Identifier to Federate with AllowCreate is set to TRUE. SP communication to the IdP_A for the SAML Authentication Request is through HTTP Redirect binding. IdP_A does not recognize User and thus can not authenticate user.

IdP_A CONFIRM: ProxyCount is set to 0.

IdP_A CONFIRM: User is not authenticated.

Step 3: Response Failure

Description: Being unable to authenticate User, IdP_A returns SAML Response with error indicating AuthnRequest failed.

SP CONFIRM: IdP_A returns SAML Response indicating authentication error.

Step 4: ProxyCount is Removed and IdP List is set

Description: SP removes ProxyCount where proxy is allowed. SP configures <IdPList> to include IdP_B.

SP CONFIRM: SP has removed ProxyCount to allow proxy.

SP CONFIRM: SP has set <IdPList> to include IdP_B.

Step 5: AuthnRequest from SP to IdP_A, Redirect Binding, Federate
Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with AllowCreate is set to TRUE. SP communication to the IdP_A for the SAML Authentication Request is through HTTP Redirect binding. IdP_A does not recognize User but recognizes it can proxy the AuthnRequest to IdP_B.

IdP_A CONFIRM: ProxyCount is not set.
IdP_A CONFIRM: User is not authenticated.
IdP_A CONFIRM: AuthnRequest contains <IdPList> which includes IdP_B.

Step 6: AuthnRequest from IdP_A to IdP_B, Redirect Binding, Federate
Description: IdP_A proxies AuthnRequest to IdP_B through HTTP Redirect binding.

IdP_B CONFIRM: Receives AuthnRequest from IdP_A.
IdP_B CONFIRM: ProxyCount is set to 0.
IdP_B CONFIRM: <IdPList> includes IdP_B.

Step 7: Assertion Response from IdP_B to IdP_A, POST binding
Description: User provides assigned credentials to IdP_B for authentication. IdP_B provides assertion of User and returns a signed SAML Response message to IdP_A through HTTP POST binding.

IdP_A CONFIRM: Receives SAML Response through HTTP POST binding.
IdP_A CONFIRM: Valid assertion is returned from IdP_B.
IdP_A CONFIRM: <AuthnStatement> contains <AuthenticatingAuthority> referencing IdP_B.

Step 8: Assertion Response from IdP_A to SP, POST binding
Description: IdP_A inserts assertion of User it received from IdP_B and returns a signed SAML Response message to SP through HTTP POST binding.

SP CONFIRM: Receives SAML Response through HTTP POST binding.
SP CONFIRM: Valid assertion is returned from IdP_A.
SP CONFIRM: <AuthnStatement> contains <AuthenticatingAuthority> referencing IdP_B.

Step 9: SLO Request, IdP-Initiated, Redirect Binding
Description: IdP_A logs out User session. IdP_A sends a signed LogoutRequest message to SP using HTTP Redirect binding.SP logs out User session. SP returns a signed LogoutResponse message to IdP_A using HTTP Redirect binding.

IdP_A CONFIRM: User logged out at IdP_A.
SP CONFIRM: Receives signed LogoutRequest through HTTP Redirect binding.
SP CONFIRM: User logged out at SP.
IdP_A CONFIRM: Receives signed LogoutResponse through HTTP Redirect binding.

Step 10: ProxyCount=1 and IdP List is set
Description: SP makes ProxyCount set to 1. SP configures <IdPList> to include IdP_B.
SP CONFIRM: SP sets ProxyCount to 1.
SP CONFIRM: SP has set <IdPList> to include IdP_B.

Step 11: AuthnRequest from SP to IdP_A, Redirect Binding, Federate
Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with AllowCreate is set to TRUE. SP communication to the IdP_A for the SAML Authentication Request is through HTTP Redirect binding. IdP_A does not recognize User but recognizes it can proxy the AuthnRequest to IdP_B.
IdP_A CONFIRM: ProxyCount is set to 1.
IdP_A CONFIRM: User is not authenticated.
IdP_A CONFIRM: AuthnRequest contains <IdPList> which includes IdP_B.

Step 12: AuthnRequest from IdP_A to IdP_B, Redirect Binding, Federate
Description: IdP_A proxies AuthnRequest to IdP_B through HTTP Redirect binding.
IdP_B CONFIRM: Receives AuthnRequest from IdP_A.
IdP_B CONFIRM: ProxyCount is set to 0.
IdP_B CONFIRM: <IdPList> includes IdP_B.

Step 13: Assertion Response from IdP_B to IdP_A, POST binding
Description: User provides assigned credentials to IdP_B for authentication. IdP_B provides assertion of User and returns a signed SAML Response message to IdP_A through HTTP POST binding.
IdP_A CONFIRM: Receives SAML Response through HTTP POST binding.
IdP_A CONFIRM: Valid assertion is returned from IdP_B.
IdP_A CONFIRM: <AuthnStatement> contains <AuthenticatingAuthority> referencing IdP_B.

Step 14: Assertion Response from IdP_A to SP, POST binding
Description: IdP_A inserts assertion of User it received from IdP_B and returns a signed SAML Response message to SP through HTTP POST binding.
SP CONFIRM: Receives SAML Response through HTTP POST binding.
SP CONFIRM: Valid assertion is returned from IdP_A.
SP CONFIRM: <AuthnStatement> contains <AuthenticatingAuthority> referencing IdP_B.

Step 15: SLO Request, IdP-Initiated, Redirect Binding
Description: IdP_A logs out User session. IdP_A sends a signed LogoutRequest message to SP using HTTP Redirect binding.SP logs out User session. SP returns a signed LogoutResponse message to IdP_A using HTTP Redirect binding.
IdP_A CONFIRM: User logged out at IdP_A.
SP CONFIRM: Receives signed LogoutRequest through HTTP Redirect binding.
SP CONFIRM: User logged out at SP.
IdP_A CONFIRM: Receives signed LogoutResponse through HTTP Redirect binding.

Test Case G – Name Identifier Mapping

Preconditions:

Metadata exchanged and loaded

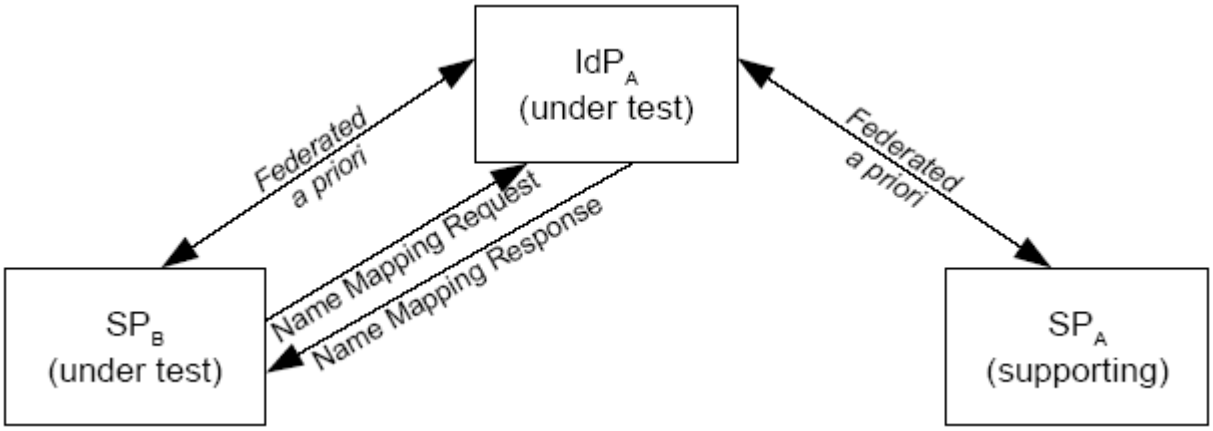
Encryption disabled

User Identities Not Federated

Conformance Modes: IdP Extended, SP Extended

Background on 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:



Step 1: SSO at SP_A

Description: User does Single Sign-On at SP_A with Persistent Name Identifier. SP_A communicates Authentication Request through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

IdP CONFIRM: SP_A successfully communicated SAML Authentication Request through HTTP Redirect binding.

IdP CONFIRM: User has been federated with SP_A.

SP_A CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

SP_A CONFIRM: User has been federated with IdP.

Step 2: SSO at SP_B

Description: User does Single Sign-On at SP_B with Persistent Name Identifier. SP_B communicates Authentication Request through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

IdP CONFIRM: SP_B successfully communicated SAML Authentication Request through HTTP Redirect binding.

IdP CONFIRM: User has been federated with SP_B.

SP_B CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

SP_B CONFIRM: User has been federated with IdP.

Step 3: NameIDMappingRequest from SP_B

13. SP_B sends signed NameIdMappingRequest message over a SOAP binding to the IdP requesting an alternative name identifier for User. IdP maps the request to the User name ID federated with SP_A. IdP returns the encrypted name ID federated with SP_A in a signed NameIdMappingResponse message using a SOAP binding.

IdP CONFIRM: Receives signed NameIdMappingRequest for name ID federated with SP_B.

SP_B CONFIRM: Receives NameIdMappingResponse for for name ID federated with SP_A.

SP_B CONFIRM: Receives Encrypted NameID.

Test Case H – IDP Introduction

Preconditions:

Metadata exchanged and loaded

Encryption disabled

User Identities Not Federated

NOTE: The SAML Conformance specification states that IdP Discovery is optional for SP and SP Lite applications. SP and SP Lite participants may option out of this test case.

Conformance Modes: IdP, SP, IdP Lite, SP Lite, eGov

Background

Two IdP actors are needed to execute this test case. Test administrator will provide specific instructions on setup and actor roles at time of test case execution.

Step 1: Clear Cookies

Description: Cookies are cleared from User Browser

USER CONFIRM: User has cleared cookies from browser.

Step 2: IdP_A is added to CDC

Description: User logins at IdP_A. Cookie is set in common domain with IdP_A appended to list of IdPs.

IdP_A CONFIRM: User logged in, cookie is set in common domain and IdP_A appended to end of IdP list in cookie.

Step 3: IdP_B is added to CDC

Description: User logins at IdP_B. IdP_B appended to list of IdPs in CDC.

IdP_B CONFIRM: User logged in and IdP_B appended to end of IdP list in CDC.

Step 4: SSO to IdP_A using CDC, HTTP Redirect

Description: User/SP does Single Sign-On using a common domain cookie. SP reads cookie. For eGov profile testing, SP must present to the User a list of IdPs and allow User to select IdP_A for authentication. For non-eGov profile testing, depending on SP implementation, either the User is presented list of IDPs and selects IdP_A for authentication or SP redirects User to IdP_A for authentication. SP communication to the IdP_A for the signed authentication request is through HTTP Redirect binding. IdP_A provides signed assertion of User and IdP returns a SAML Response message through HTTP POST binding.

IdP_A CONFIRM: SP successfully communicated signed SAML Authentication Request through HTTP Redirect binding.

SP CONFIRM: Cookie was read and IdP_A and IdP_B were present in CDC.

SP CONFIRM: IdP_A returns signed assertion through HTTP POST binding.

SP CONFIRM: For eGov profile, SP presents list of IdPs for authentication and IdP_A and IdP_B must be present on list.

Step 5: SLO, SP-Initiated, HTTP Redirect

Description: SP does SLO. SP sends a signed LogoutRequest message to IdP_A using HTTP Redirect binding. IdP_A returns a signed LogoutResponse message. User is logged out.

IdP_A CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

SP CONFIRM: User is logged out.

Step 6: CDC is removed

Description: User closes browser. CDC is removed.

User CONFIRM: CDC is removed once browser is closed.

Test Case I – Single Session Logout

Preconditions:

Metadata exchanged and loaded
Encryption disabled
User Identities Not Federated

Conformance Modes: IdP, SP, IdP Lite, SP Lite, eGov

Step 1: SSO creates Session A for User

Description: User creates Session A through Single Sign-On with Federate where AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
IdP CONFIRM: User has been federated.
IdP CONFIRM: User has been logged in through Session A.
SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

Step 2: SSO creates Session B for User

Description: User creates new Session B, generally through second browser instances, through Single Sign-On. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
IdP CONFIRM: User has been logged in through Session B.
SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

Step 3: SLO from SP for Session A

Description: User logs off of Session A at the SP. SP sends a signed LogoutRequest message to IdP for Session A using HTTP Redirect binding. IdP examines <SessionIndex> and determins the logout request is for Session A. User is logged out of Session A, but User remains logged in through Session B. IdP returns a signed LogoutResponse message for Session A.

IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
IdP CONFIRM: User logged out of Session A.
IdP CONFIRM: User remains logged in through Session B.
SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.
SP CONFIRM: User logged out of Session A.
SP CONFIRM: User remains logged in through Session B.

Step 4: SSO creates Session C for User

Description: User creates Session C through Single Sign-On with Federate where AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.

IdP CONFIRM: User has been federated.

IdP CONFIRM: User has been logged in through Session C.

SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

Step 5: SLO from IdP for Session C

Description: User logs off of Session C at the IdP. IdP sends a signed LogoutRequest message to SP for Session C using HTTP Redirect binding. SP examines <SessionIndex> and determins the logout request is for Session C. User is logged out of Session C, but User remains logged in through Session B. SP returns a signed LogoutResponse message for Session C.

SP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

SP CONFIRM: User logged out of Session C.

SP CONFIRM: User remains logged in through Session B.

IdP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

IdP CONFIRM: User logged out of Session C.

IdP CONFIRM: User remains logged in through Session B.

Test Case J – Unsolicited <Response> and "Transient" NameID

Preconditions:

Metadata exchanged and loaded
Encryption disabled
User Identities Not Federated

Conformance Modes: IdP, SP, IdP Lite, SP Lite, eGov

Step 1: Unsolicited <Response>, HTTP Post Binding, "transient" NameID
Description: User does Single Sign-On at IdP. IdP provides assertion of User and makes Name ID format "transient". IdP sends a signed SAML Response message through HTTP POST binding.

IdP CONFIRM: User has been federated.
SP CONFIRM: NameID format is "transient".
SP CONFIRM: IdP sends signed SAML Response through HTTP POST binding.

Step 2: SLO from SP
Description: SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP logs out User session. IdP returns a signed LogoutResponse message.

IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

Step 3: Unsolicited <Response>, Artifact Binding, "transient" NameID
Description: User does Single Sign-On at IdP. IdP provides assertion of User and makes Name ID is format "transient". <Response> message is communicated through Artifact binding. The IdP and SP resolve the artifact via a SOAP binding. SP consumes the <Response> message.

IdP CONFIRM: Artifact resolution is properly done.
IdP CONFIRM: User has been federated
SP CONFIRM: NameID format is "transient".
SP CONFIRM: IdP sends signed SAML Response through HTTP Artifact.
SP CONFIRM: Artifact resolution is properly done.

Step 4: SLO from IdP
Description: IdP sends a signed LogoutRequest message to SP using HTTP Redirect binding. SP logs out User session. SP returns a signed LogoutResponse message.

SP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
IdP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

Test Case K – Multiple SP Logout

Preconditions:

Metadata exchanged and loaded

Encryption disabled

User Identities Not Federated

Conformance Modes: IdP, SP, IdP Lite, SP Lite, eGov

Step 1: SSO from SP_A

Description: User at SP_A performs Single Sign-On (any profile) to IdP.

IdP CONFIRM: SP_A successfully communicated SAML Authentication Request and IdP sent back Assertion for User.

IdP CONFIRM: User has been federated with SP_A

SP_A CONFIRM: IdP returns signed SAML Response and User is authenticated.

Step 2: SSO from SP_B using same Session ID

Description: User logins to SP_B and is authenticated by IdP with same session id.

IdP CONFIRM: SP_B successfully communicated SAML Authentication Request and IdP sent back Assertion for User and maintained same session id as in Step 1.

IdP CONFIRM: User has been federated with SP_B

SP_B CONFIRM: IdP returns signed SAML Response and User is authenticated.

Step 3: SLO from SP_A to IdP

Description: User issues SLO from SP_A to IdP.

IdP CONFIRM: SP_A sends signed LogoutRequest for User.

SP_A CONFIRM: A signed LogoutRequest is sent to IdP.

Step 4: LogoutRequest from IdP to SP_B

Description: Signed LogoutRequest is sent from IdP to SP_B. User is logged out of SP_B. After receiving the LogoutResponse from SP_B, IdP sends LogoutResponse to SP_A.

IdP CONFIRM: Signed LogoutRequest is sent to SP_A and receives back signed LogoutResponse.

IdP CONFIRM: No active session for User.

SP_B CONFIRM: IdP sends signed LogoutResponse, a signed LogoutResponse is returned and User is logged out.

SP_A CONFIRM: Receives signed LogoutResponse from IdP.

Step 5: SSO from SP_B to IdP

Description: User at SP_B performs Single Sign-On (any profile) to IdP.

IdP CONFIRM: SP_B successfully communicated SAML Authentication Request and IdP sent back Assertion for User.

IdP CONFIRM: User has active session.

SP_B CONFIRM: IdP returns signed SAML Response and User is authenticated.

968 **Step 6: SSO from SP_A using same Session ID**
969 Description: User logins to SP_A and is authenticated by IdP with same session id.
970 IdP CONFIRM: SP_A successfully communicated SAML Authentication Request and IdP sent
971 back Assertion for User and maintained same session id as in Step 5.
972 SP_A CONFIRM: IdP returns signed SAML Response and User is authenticated.

973 **Step 7: SLO from SP_B to IdP**
974 Description: User does SLO from IdP to SP_B.
975 IdP CONFIRM: SP_B is sent signed LogoutRequest for User.
976 SP_B CONFIRM: IdP sends a signed LogoutRequest and User is logged out.

977 **Step 8: LogoutRequest from IdP to SP_A**
978 Description: Signed LogoutRequest is sent to SP_A from IdP. User is logged out of SP_A. After
979 receiving the LogoutResponse from SP_A, IdP sends LogoutResponse to SP_B.
980 IdP CONFIRM: Signed LogoutRequest is sent to SP_A and receives back signed
981 LogoutResponse.
982 SP_A CONFIRM: IdP sends signed LogoutResponse, a signed LogoutResponse is returned
983 and User is logged out.
984 SP CONFIRM: Receives signed LogoutResponse from IdP.

985 **Step 9: SSO from SP_B to IdP**
986 Description: User at SP_B performs Single Sign-On (any profile) to IdP.
987 IdP CONFIRM: SP_B successfully communicated SAML Authentication Request and IdP sent
988 back Assertion for User.
989 IdP CONFIRM: User has active session.
990 SP_B CONFIRM: IdP returns signed SAML Response and User is authenticated.

991 **Step 10: SSO from SP_A using same Session ID**
992 Description: User logins to SP_A and is authenticated by IdP with same session id.
993 IdP CONFIRM: SP_A successfully communicated SAML Authentication Request and IdP sent
994 back Assertion for User and maintained same session id as in Step 5.
995 SP_A CONFIRM: IdP returns signed SAML Response and User is authenticated.

Step 11: Local logout at SP_B

Description: User does local logout (not SLO) at SP_B.

IdP CONFIRM: LogoutRequest for User is not received at this time.

SP_B CONFIRM: User is logged out locally.

Step 12: SLO from SP_A to IdP

Description: User issues SLO from SP_A to IdP.

IdP CONFIRM: SP_A sends signed LogoutRequest for User.

SP_A CONFIRM: A signed LogoutRequest is sent to IdP. User is logged out.

Step 13: PartialLogout Error

Description: Signed LogoutRequest is sent from IdP to SP_B. Because User is already logged out of SP_B, a status code of "PartialLogout" is returned in the to the Signed LogoutResponse. IdP sends LogoutResponse to SP_A.

IdP CONFIRM: Signed LogoutRequest is sent to SP_B and receives back signed LogoutResponse.

IdP CONFIRM: Signed LogoutReponse contains status code of urn:oasis:names:tc:SAML:2.0:status:PartialLogout.

SP_B CONFIRM: IdP sends signed LogoutResponse, unable to perform SLO, and a signed LogoutResponse is returned indicating "PartialLogout".

SP_A CONFIRM: Receives signed LogoutResponse from IdP indicating "PartialLogout."

Test Case L – Force Authentication and Passive Authentication

Preconditions:

Metadata exchanged and loaded

Encryption disabled

Conformance Modes (Required): IdP, SP, IdP Lite, SP Lite, eGov

Step 1: User Logins at IdP

Description: User logins at IdP and creates and active session

IdP CONFIRM: User logged in.

Step 2: SP sets IsPassive=TRUE

Description: SP is configured to make IsPassive set to TRUE.

SP CONFIRM: SP is configured IsPassive=TRUE.

Step 3: SSO with isPassive=TRUE

Description: User/SP does Single Sign-On SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding. IdP provides assertion of User without interacting with the user. IdP returns a signed SAML Response message through HTTP POST binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.

IdP CONFIRM: User does not interact with IdP or IdP must not take control of user interface.

SP CONFIRM: IdP returns assertion in signed SAML Response through HTTP POST binding.

Step 4: SLO from SP

Description: SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP logs out User session. IdP returns a signed LogoutResponse message.

IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

SP CONFIRM: User is logged out.

Step 5: SP sets IsPassive=FALSE

Description: SP is configured to make IsPassive set to FALSE.

SP CONFIRM: SP is configured IsPassive=FALSE.

Step 6: SSO with isPassive=FALSE

Description: User/SP does Single Sign-On SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding. IdP interacts with and authenticates the user. IdP returns a signed SAML Response message through HTTP POST binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
IdP CONFIRM: User does interact with IdP.
SP CONFIRM: IdP returns assertion in signed SAML Response through HTTP POST binding.

Step 7: SLO from SP
Description: SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP logs out User session. IdP returns a signed LogoutResponse message.
IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.
SP CONFIRM: User is logged out.

Step 8: User Logins At IdP
Description: User logins at IdP and creates and active session
IdP CONFIRM: User logged in.

Step 9: SP sets ForceAuthn=TRUE
Description: SP is configured to make ForceAuthn set to TRUE.
SP CONFIRM: SP is configured ForceAuthn=TRUE.

Step 10: SSO with ForceAuthn=TRUE
Description: User/SP does Single Sign-On SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding. IdP interacts with User and authenticates the User. IdP provides assertion of User. IdP returns a signed SAML Response message through HTTP POST binding.
IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.
IdP CONFIRM: User interacts with IdP and is authenticated.
SP CONFIRM: IdP returns assertion in signed SAML Response through HTTP POST binding.

Step 11: SLO from SP
Description: SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP logs out User session. IdP returns a signed LogoutResponse message.
IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.
SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.
SP CONFIRM: User is logged out.

Test Case M – SAML Authentication Authority

Preconditions:

Metadata exchanged and loaded
Encryption disabled
User Identities Not Federated

Conformance Modes: SAML Authentication Authority

Note: Section [AuthenticationContexts] within this document describes the strength of the AuthnContext classes used for comparison.

Test Steps

Step 1:
Description: User/SP does Single Sign-On with Persistent Name Identifier. SP communication to the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP POST binding.
IdP CONFIRM: User has been federated
SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

Step 2:
Description: SAML Requester sets AC comparison to “exact”.

SAML Requester CONFIRM: AC comparison=”exact”.

Step 3:
Description: SAML Requester sends Authentication Query to SAML Responder through SOAP binding. SAML Responder returns SAML Response.

SAML Responder CONFIRM: SAML Requester sent Authentication Query.
SAML Requester CONFIRM: SAML Responder returned the SAML Response.

Step 4:
Description: SAML Requester sets AC comparison to “better”.

SAML Requester CONFIRM: AC comparison=”better”.

Step 5:
Description: SAML Requester sends Authentication Query to SAML Responder through SOAP binding. SAML Responder returns SAML Response.

SAML Responder CONFIRM: SAML Requester sent Authentication Query.
SAML Requester CONFIRM: SAML Responder returned the SAML Response.

Step 6:
Description: SAML Requester sets AC comparison to “minimum”.

SAML Requester CONFIRM: AC comparison=”minimum”.

Step 7:
Description: SAML Requester sends Authentication Query to SAML Responder through SOAP binding. SAML Responder returns SAML Response.
SAML Responder CONFIRM: SAML Requester sent Authentication Query.
SAML Requester CONFIRM: SAML Responder returned the SAML Response.

Step 8:
Description: SAML Requester sets AC comparison to “maximum”.
SAML Requester CONFIRM: AC comparison=” maximum”.

Step 9:
Description: SAML Requester sends Authentication Query to SAML Responder through SOAP binding. SAML Responder returns SAML Response.
SAML Responder CONFIRM: SAML Requester sent Authentication Query.
SAML Requester CONFIRM: SAML Responder returned the SAML Response.

Test Case N – SAML Attribute Authority

Preconditions:

Metadata exchanged and loaded
Encryption disabled
User Identities Not Federated

Conformance Modes: SAML Attribute Authority

Step 1:
Description: User/SP does Single Sign-On with Persistent Name Identifier. SP communication to the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.
IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP POST binding.
IdP CONFIRM: User has been federated
SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

Step 2:
Description: SAML Responder sets attribute query to no attributes.
SAML Responder CONFIRM: Attribute Query No Attributes.

Step 3:
Description: SAML Requester sends Attribute Query to SAML Responder through SOAP binding. SAML Responder returns SAML Response.
SAML Responder CONFIRM: SAML Requester sent Attribute Query.
SAML Requester CONFIRM: SAML Responder returned the SAML Response.

Step 4:
Description: SAML Responder sets attribute query to attribute named.
SAML Responder CONFIRM: Attribute Query Attribute Named.

Step 5:
Description: SAML Requester sends Attribute Query to SAML Responder through SOAP binding. SAML Responder returns SAML Response.
SAML Responder CONFIRM: SAML Requester sent Attribute Query.
SAML Requester CONFIRM: SAML Responder returned the SAML Response.

Step 6:
Description: SAML Responder sets attribute query to attribute value.
SAML Responder CONFIRM: Attribute Query Attribute Value.

Step 7:
Description: SAML Requester sends Attribute Query to SAML Responder through SOAP binding. SAML Responder returns SAML Response.

SAML Responder CONFIRM: SAML Requester sent Attribute Query.
SAML Requester CONFIRM: SAML Responder returned the SAML Response.

Step 8:
Description: SAML Responder sets attribute query to attribute named. SAML Responder enables attribute for encryption.
SAML Responder CONFIRM: Attribute Query Attribute Named.
SAML Responder CONFIRM: Encryption assertion enabled.

Step 9:
Description: SAML Requester sends Attribute Query to SAML Responder through SOAP binding. SAML Responder returns SAML Response.
SAML Responder CONFIRM: SAML Requester sent Attribute Query.
SAML Requester CONFIRM: SAML Responder returned the SAML Response.

Test Case O – SAML Authorization Decision Authority

Preconditions:

Metadata exchanged and loaded

Encryption disabled

User Identities Not Federated

Conformance Modes: SAML Authorization Decision Authority

Step 1:

Description: User/SP does Single Sign-On with Persistent Name Identifier. SP communication to the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP POST binding.

IdP CONFIRM: User has been federated

SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

Step 2:

Description: SAML Requester enables HTTP Basic Authentication.

SAML Requester CONFIRM: HTTP Basic Authentication enabled.

Step 3:

Description: SAML Responder sets Authorization Query to never permitted which means subject is never authorized for access.

SAML Responder CONFIRM: AuthzQuery Resource=never

Step 4:

Description: SAML Requester sends Authorization Query to SAML Responder through SOAP binding. SAML Responder returns SAML Response.

SAML Responder CONFIRM: SAML Requester sent Authorization Query.

SAML Requester CONFIRM: SAML Responder returned the SAML Response.

Step 5:
Description: SAML Responder sets authorization query to maybe permitted if authentication is matched which means subject is authorized if it is a “particular” subject.
SAML Responder CONFIRM: AuthzQuery Resource=maybe

Step 6:
Description: SAML Requester sends Authorization Query to SAML Responder through SOAP binding. SAML Responder returns SAML Response.
SAML Responder CONFIRM: SAML Requester sent Authorization Query.
SAML Requester CONFIRM: SAML Responder returned the SAML Response.

Step 7:
Description: SAML Responder sets Authorization Query to always permitted which means subject is always authorized.
SAML Responder CONFIRM: AuthzQuery Resource=always

Step 8:
Description: SAML Requester sends Authorization Query to SAML Responder through SOAP binding. SAML Responder returns SAML Response.
SAML Responder CONFIRM: SAML Requester sent Authorization Query.
SAML Requester CONFIRM: SAML Responder returned the SAML Response.

Test Case P – Error Testing

Preconditions:

Metadata exchanged and loaded
Encryption disabled
User Identities Not Federated

Conformance Modes: IdP, SP, SP Lite, eGov, POST

NOTE – Test Steps 2-11 involve the Liberty Error Test Tool. Metadata for conducting these tests will be exchanged.

Step 1:
Description: Successful SSO using Artifact Resolution as described in Steps 1-3 of Test Case B are done. Once those steps are complete, the SP reissues the same <Artifact> in a new <ArtifactResolve> message. The IdP recognizes the reissued <Artifact> and refuses it. <ArtifactResponse> is returned with no embedded message.

IdP CONFIRM: Successful SSO using Artifact Binding.
IdP CONFIRM: Second <ArtifactResolve> message received using same <Artifact> and refused.
SP CONFIRM: <ArtifactResponse> is returned with no embedded message.

Step 2:
Description: Test Harness POSTs an unsolicited SAML Response message containing a valid assertion.

SP CONFIRM: SAML Response was received and assertion accepted.

Step 3:
Description: Test Harness re-POSTs the assertion that was successful during the initialization of this test sequence.

SP CONFIRM: Assertions are not replayed within the validity period of the assertion.

Step 4:
Description: The assertion of the SAML Response from Step 2 is altered and sent without re-signing in a HTTP POST from Test Harness.

SP CONFIRM: SP rejects the message.

Step 5:
Description: The assertion of the SAML Response from Step 2 is sent but signed with the wrong signing key in a HTTP POST from Test Harness.

SP CONFIRM: SP rejects the message.

Step 6:
Description: The Test Harness constructs a SAML Response message with an incorrect Recipient attribute. Recipient attribute is in the <SubjectConfirmationData> element.
SP CONFIRM: SP detects and rejects the message.

Step 7:
Description: The Test Harness sends an altered assertion in the SAML Response. A different Method URN is substituted in the assertion's <SubjectConfirmation> element other than the required Method of urn:oasis:names:tc:SAML:2.0:cm:bearer.
SP CONFIRM: SP detects and rejects the message.

Step 8:
Description: The Test Harness POSTs a SAML Response containing an assertion which does not contain an <AudienceRestriction> including the SP's unique identifier as an <Audience>.
SP CONFIRM: SP rejects the assertion.

Step 9:
Description: The Test Harness sets the *NotOnOrAfter* attribute to a date/time that has occurred in past with respect the date/time of executing this test step.
SP CONFIRM: The SP to reject the assertion because of the *NotOnOrAfter* attribute.

Step 10:
Description: The Test Harness sets the *NotBefore* attribute to a date/time in the future with respect to the date/time of executing this test step.
SP CONFIRM: The SP to reject the assertion because of the *NotBefore* attribute.

Step 11:
Description: The Test Harness includes a <Condition> extension element in the <Conditions> element of the assertion that cannot be understood.
SP CONFIRM: The SP rejects the assertion.

Test Case Q – Requested AuthnContext

Preconditions:

Metadata exchanged and loaded
Encryption disabled
User Identities Not Federated

Conformance Modes: eGov Profile
Note: Section [AuthenticationContexts] within this document describes the strength of the AuthnContext classes used for comparison used in this test case.

Step 1: Issue <AuthnRequest> with Assigned <RequestedAuthnContext>
Description: For each iteration in Table Q.1, SP sends an <AuthnRequest> to the IdP. Within <NameIDPolicy>, AllowCreate is set to "true", and the with format is set to 'persistent'. The *ForceAuthn* attribute is set to "true". SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding.

For each iteration, the SP inserts a <RequestedAuthnContext> element into the <AuthnRequest> message. The authentication context requested and the *Comparison* attribute setting is defined in Table Q.1. Prior to each iteration, the IdP enables its authenticating context for the User as defined in the table. The expected Status value for the <Response> message is also listed in the table.

TABLE Q.1

Iteration	SP Requested AC	*Comparison*	IdP Supported AC	Status Response
1	Password	"exact"	InternetProtocol	NoAuthnContext
2	Password	"minimum"	InternetProtocol	NoAuthnContext
3	Password	"better"	InternetProtocol	NoAuthnContext
4	InternetProtocol	"exact"	InternetProtocol	Success
5	InternetProtocol	"minimum"	InternetProtocol	Success
6	InternetProtocol	"maximum"	InternetProtocol	Success
7	InternetProtocol	"maximum"	Password	NoAuthnContext
8	InternetProtocol	"better"	Password	Success

SP CONFIRM: Every iteration from Table Q.1 is executed, and all messages, actions and responses match the results assigned by the table.
IdP CONFIRM: Every iteration from Table Q.1 is executed, and all messages, actions and responses match the results assigned by the table.

Test Case R – User Consent

Preconditions:

Metadata exchanged and loaded
Encryption disabled
User Identities Not Federated

Conformance Modes: eGov

Step 1: User Consent StatusResponse
Description: IdP must provide means for User to provide authentication consent per the different consent values listed in Table R.1. Consent conditions are listed in section 8.4 of [SAMLCore]. The exact means used is left to the individual IdP. After user provides assigned credentials for authentication, IdP provides assertion of User and returns <Assertion> in an unsolicited signed SAML Response message through HTTP POST binding. The *Consent* attribute is included in the StatusResponse. The test step is repeated through each iteration in Table R.1

TABLE R.1

Iteration	Consent value
1	urn:oasis:names:tc:SAML:2.0:consent:obtained
2	urn:oasis:names:tc:SAML:2.0:consent:prior
3	urn:oasis:names:tc:SAML:2.0:consent:current-implicit
4	urn:oasis:names:tc:SAML:2.0:consent:current-explicit
5	urn:oasis:names:tc:SAML:2.0:consent:unspecified

SP CONFIRM: IdP sends signed SAML Response through HTTP POST binding.
SP CONFIRM: Valid assertion is returned from IdP.
SP CONFIRM: *Consent* attribute match values in Table R.1
SP CONFIRM: User A identity has been federated with IdP.
IdP CONFIRM: User A identity has been federated with SP.

Test Case S – Assertion Attribute

Preconditions:

Metadata exchanged and loaded
Encryption disabled
User Identities Not Federated

Conformance Modes: eGov

Step 1: User A, AttributeStatement in Assertion Response
Description: User A requires authentication. SP sends <AuthnRequest> with AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding. User A provides assigned credentials for authentication. IdP provides assertion of User A. The attributes in the table below are assigned to User A and are to be returned in a single <AttributeStatement> in the assertion. IdP returns <Assertion> in a signed SAML Response message through HTTP POST binding.

TABLE S.1

Attribute Name	AttributeValue (string)	NameFormat
LastName	Wall	“basic”
urn:oid:2.5.4.40	John	“uri”
Position	PG	“unspecified”

SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.
SP CONFIRM: Valid assertion is returned from IdP.
SP CONFIRM: Returned attributes match values in Table S.1
SP CONFIRM: User A identity has been federated with IdP.
IdP CONFIRM: User A identity has been federated with SP.

Step 2: User B, No AttributeStatement in Assertion Response
Description: User B requires authentication. SP sends <AuthnRequest> with AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding. User B provides assigned credentials for authentication. IdP provides assertion of User B. No <AttributeStatement> is returned in the <Assertion>.

SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.
SP CONFIRM: Valid assertion is returned from IdP.
SP CONFIRM: No <AttributeStatement> is returned in <Assertion>.
SP CONFIRM: User B identity has been federated with IdP.
IdP CONFIRM: User B identity has been federated with SP.

Test Case T – Unspecified Format

Preconditions:

Metadata exchanged and loaded

Encryption disabled

User Identities Not Federated

Conformance Modes: eGov

Step 1: AuthnRequest, 'Unspecified' NameID format, Redirect Binding, Federate

Description: User/SP does Single Sign-On with AllowCreate is set to TRUE. The with Name Identifier format is set to 'unspecified'. SP communication to the IdP for the SAML Authentication Request is through HTTP Redirect binding.

IdP CONFIRM: SP successfully communicated SAML Authentication Request through HTTP Redirect binding.

IdP CONFIRM: Name ID format is 'unspecified'.

Step 2: Assertion Response, POST binding

Description: User provides assigned credentials for authentication. IdP provides assertion of User. NameID format is set to 'persistent'. In <Assertion>, *SessionIndex* attribute must be present but *SessionNotOnOrAfter* must not be present. IdP returns <Assertion> in a signed SAML Response message through HTTP POST binding.

SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

SP CONFIRM: Valid assertion is returned from IdP.

SP CONFIRM: NameID format is 'persistent'.

SP CONFIRM: *SessionIndex* is present.

SP CONFIRM: *SessionNotOnOrAfter* is not present.

SP CONFIRM: User identity has been federated with IdP.

IdP CONFIRM: User identity has been federated with SP.

References

[SAMLTP31] Kyle Meadors, et al, "SAML 2.0 Interoperability Testing Procedures, V3.1, Errata J," Liberty Alliance Project (July 2008), http://www.projectliberty.org/liberty/content/download/4160/27946/file/Liberty_Interoperability_SAML_Test_Plan_v3.1.pdf

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

[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

[SAMLLDAP] S. Cantor et al, "SAML V2.0 X.500/LDAP Attribute Profile," OASIS SSTC (December 19, 2006), http://docs.oasis-open.org/security/saml/SpecDrafts-Post2.0/sstc-saml-attribute-x500-cd-01.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