Abstract

This document specifies a SAMLv2 lightweight Web Browser Single Sign-On Profile. This profile is modeled on the OASIS SAMLv2 Web Browser SSO profile, adding various constraints, and using a new lightweight SAMLv2 HTTP POST binding which does not rely on XML Digital Signature -- relying on a more simple-to-implement signature approach instead.
Table of Contents

1. Introduction ................................................. 3
2. Terminology .................................................. 4
3. Specification Scope ......................................... 5
4. SAML Introduction ........................................... 6
  4.1. SAML Assertions ........................................... 7
  4.2. Abstract Request/Response Protocol ....................... 8
5. Lightweight SAML Profiles .................................... 9
  5.1. Lightweight Web Browser SSO SAML Profile ................. 9
    5.1.1. Differences from SAMLv2 Web Browser SSO Profile ....... 9
    5.1.2. Required Information .................................. 10
    5.1.3. Profile Overview ..................................... 10
    5.1.4. Profile Description ................................... 14
    5.1.5. Use of Authentication Request Protocol ............... 16
    5.1.6. Unsolicited Responses ................................ 19
    5.1.7. Use of Metadata ...................................... 20
  5.2. Example SAML Assertion ................................... 20
  5.3. Security Considerations ................................. 21
    5.3.1. Unintended Principal Data Leakage .................... 21
    5.3.2. Man-in-the-middle Attacks and Stolen Assertions ....... 22
    5.3.3. Forged Assertion ..................................... 22
  5.4. Contributors ............................................. 23
  5.5. Acknowledgments ......................................... 23
  5.6. IANA Considerations ..................................... 23
  5.7. Open Issues .............................................. 23
6. References .................................................... 24
  6.1. Normative References .................................... 24
  6.2. Informative References .................................. 25
Authors’ Addresses ............................................. 27
Intellectual Property and Copyright Statements ..................... 28
1. Introduction

This document specifies a SAMLv2 lightweight Web Browser Single Sign-On Profile. This profile is modeled on the OASIS SAMLv2 Web Browser SSO profile, adding various constraints, and using a new lightweight SAMLv2 HTTP POST binding which does not rely on XML Digital Signature -- relying on a more simple-to-implement signature approach instead.

XML digital signature (XMLdsig) [W3C.xmldsig-core] is not used at all because it is asserted that implementation support for it is essentially non-existent in so-called "scripting" environments, e.g. PERL/PYTHON/PHP/Ruby, and/or different implementations of it are not terribly interoperable as yet, due to the inherent complexity of the specification and its required behaviors.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].
3. Specification Scope

The scope of this specification is:

- Specify a "lightweight" SAML "web browser single sign-on" profile.

The following are outside the scope of this specification:

- Mechanisms for evaluating keys or certificates used for signing.
4. SAML Introduction

SAML [OASIS.sstc-saml-exec-overview-2.0-cd-01] [OASIS.sstc-saml-tech-overview-2.0-draft-08] defines an XML-based framework for exchanging "security assertions" between entities. In the course of making, or relying upon such assertions, SAML system entities may use SAML protocols, or other protocols, to communicate regarding an assertion itself, or the subject of an assertion.

Thus one can employ SAML to encode statements such as "Alice has these profile attributes and her domain's certificate is available over there, and I’m making this statement, and here’s who I am." Then one can cause such an assertion to be conveyed to some party who can then rely on it in some fashion for some purpose, for example input it into some local policy evaluation for access to some resource. This is done in a particular "context of use". Such a context of use could be, for example, deciding whether to accept and act upon a SIP-based invitation to initiate a communication session.

The specification of how SAML is employed in a particular context of use is known as a "SAML profile". The specification of how SAML assertions and/or protocol messages are concretely conveyed in, or over, another protocol is known as a "SAML Binding". Typically, a SAML profile specifies the SAML binding(s) that are to be used in its context. Specification of both or either SAML profiles and SAML bindings are by definition built on the foundation provided by the SAML specifications, especially the SAML Assertions and Protocols, aka "SAML Core", specification [OASIS.saml-core-2.0-os].

There is an additional subtle aspect of SAML profiles that is worth highlighting -- the notion of a "SAML assertion profile". A SAML assertion profile is the specification of the assertion contents in the context of a particular SAML profile. It is possibly further qualified by a particular implementation and/or deployment context. Condensed examples of SAML assertion profiles are:

- The SAML assertion must contain at least one authentication statement and no other statements. The relying party must be represented in the <AudienceRestriction> element. The SubjectConfirmation Method must be Foo. etc.

- The SAML assertion must contain at least one attribute statement and may contain more than one. The values for the subject’s profile attributes named "Foo" and "Bar" must be present. An authentication statement may be present. etc.

The primary facets of SAML itself are:
o Assertions

o Abstract Request/Response protocol

We describe each in turn below:

4.1. SAML Assertions

A SAML assertion is a package of information including issuer and subject, conditions and advice, and/or attribute statements, and/or authentication statements and/or other statements. Statements may or may not be present. The SAML assertion "container" itself contains the following information:

Issuing information:

Who issued the assertion, when was it issued and the assertion identifier.

Subject information:

The name of the subject, the security domain and optional subject information, like public key.

Conditions under which the assertion is valid:

Special kind of conditions like assertion validity period, audience restriction and target restriction.

Additional advice:

Explaining how the assertion was made, for example.

In terms of SAML assertions containing SAML attribute statements or SAML authentication statements, here are explanatory examples:

With a SAML assertion containing a SAML attribute statement, an issuing authority is asserting that the subject is associated with certain attributes with certain subject profile attribute values. For example, user jon@cs.example.com is associated with the attribute "Department", which has the value "Computer Science".

With a SAML assertion containing a SAML authentication statement, an issuing authority is asserting that the subject was authenticated by certain means at a certain time.
With a SAML assertion containing both a SAML attribute statement and a SAML authentication statement, an issuing authority is asserting the union of the above.

### 4.2. Abstract Request/Response Protocol

SAML defines an abstract request/response protocol for obtaining assertions. See Section 3 "SAML Protocols" of [OASIS.saml-core-2.0-os]. A request asks for an assertion. A response returns the requested assertion or an error. This abstract protocol may then be cast into particular contexts of use by binding it to specific underlying protocols, e.g., HTTP or SIP, and "profiling" it for the specific use case at hand. The SAML HTTP-based web single sign-on profile is one such example (see Section 4.1 Web Browser SSO Profile of [OASIS.saml-profiles-2.0-os]).
5. Lightweight SAML Profiles

This section defines one SAML profile:

The "Lightweight Web Browser SSO SAML Profile"

5.1. Lightweight Web Browser SSO SAML Profile

In the scenario supported by the web browser SSO profile, a web user either accesses a resource at a service provider, or accesses an identity provider such that the service provider and desired resource are understood or implicit. The web user authenticates (or has already authenticated) to the identity provider, which then produces an authentication assertion (possibly with input from the service provider) and the service provider consumes the assertion to establish a security context for the web user. During this process, a name identifier might also be established between the providers for the principal, subject to the parameters of the interaction and the consent of the parties.

To implement this scenario, a profile of the SAML Authentication Request protocol is used \[OASIS.saml-core-2.0-os\], in conjunction with the HTTP POST-NoXMLdsig binding \[OASIS.draft-hodges-saml-binding-noxmldsig-01\].

It is assumed that the user is using a standard commercial browser and can authenticate to the identity provider by some means outside the scope of SAML.

5.1.1. Differences from SAMLv2 Web Browser SSO Profile

This profile is similar to the Web Browser SSO Profile in \[OASIS.saml-profiles-2.0-os\]. Below we list the most salient differences between them, from the perspective of this profile:

Employs the HTTP POST-NoXMLdsig SAML binding, rather than the other bindings specified in \[OASIS.saml-bindings-2.0-os\], thus removing the binding-layer dependency on XMLdsig \[W3C.xmldsig-core\]. Also, this profile does not otherwise reference \[W3C.xmldsig-core\] from the profile itself.

Various simplifications to the option settings in the <AuthnRequest> message:

AllowCreate is always "true".

<Subject> and <Conditions> are disallowed.
Only a single assertion is returned.

Requirements for the identity provider to validate the service provider’s assertion consumer service are relaxed.

5.1.2. Required Information

The information given in this section is similar to the info provided when registering something, a MIME Media Type, say, with IANA. In this case, it is for registering this profile with the OASIS SSTC. See Section 2 "Specification of Additional Profiles" in [OASIS.saml-profiles-2.0-os].

Identification:

urn:ietf:params:?:?

@@ NOTE: This URN must be agreed upon, and then registered with IANA per [RFC3553].

Contact Information:

@@ someone’s or something’s contact info goes here

SAML Confirmation Method Identifiers:

@@ note which ones we use in text below here

Description:

Given below.

Updates:

None.

5.1.3. Profile Overview

Figure 1 illustrates this profile’s overall protocol flow. The following steps correspond to the labeled interactions in the figure. Within an individual step, there may be one or more actual message exchanges depending upon the protocol binding employed for that particular step and other implementation-dependent behavior.
1. User Agent attempts to access some resource at the Service Provider

[Do I have a security context for this UA? Hm, no, so I’m going to establish one..]

2. Service Provider determines Identity Provider to use (methods vary, details not shown)

3. `<AuthnRequest>` msg issued by Service Provider to Identity Provider

   +--------+ (redirect to IDP)
   |        |
   |        +------------------------>

4. Identity Provider identifies Principal (methods vary, details not shown)

   <=--------------------------------------------->

5. `<Response>` message issued by Identity Provider to Service Provider

   +---------------------------------------------+ (redirect to SP)
   |                                          |
   | +----------------------------------------->
   |                                           |
   |                                           |
   |                                           |
6. Based on the Identity Provider identifying (or not) the Principal, the Service Provider either returns the resource or an (HTTP) error

Figure 1: Basic flow for achieving SSO
Step 1. HTTP Request to Service Provider

In step 1, the principal, via an HTTP User Agent, makes an HTTP request for a secured resource at the service provider without a security context.

Step 2. Service Provider Determines Identity Provider

In step 2, the service provider obtains the location of an endpoint at an identity provider for the authentication request protocol that supports its preferred binding. The means by which this is accomplished is implementation-dependent. The service provider MAY use the SAML identity provider discovery profile described in [OASIS.saml-profiles-2.0-os] or any other means of identity provider discovery.

Step 3. <AuthnRequest> issued by Service Provider to Identity Provider

In step 3, the service provider issues an <AuthnRequest> message to be delivered by the user agent to the identity provider. The HTTP POST-NoXMLdsig binding MUST be used to transfer the message to the identity provider through the user agent.

Step 4. Identity Provider identifies Principal

In step 4, the principal is identified by the identity provider by some means outside the scope of this profile. This may require a new act of authentication, or it may reuse an existing authenticated session, where said authenticated session state is maintained in some unspecified (herein) manner. A common means is for the user agent to maintain session state local to the identity provider via the means of "cookies" [RFC2965].

Step 5. Identity Provider issues <Response> to Service Provider

In step 5, the identity provider issues a <Response> message to be delivered by the user agent to the service provider. The HTTP POST-NoXMLdsig binding MUST be used to transfer the message to the service provider through the user agent. The message may indicate an error, or will include (at least) an authentication assertion.
Step 6. Service Provider grants or denies access to Principal

In step 6, having received the response from the identity provider, the service provider can respond to the principal’s user agent with its own error, or can establish its own security context for the principal and return the requested resource.

Note that an identity provider can initiate this profile at step 5 and issue a <Response> message to a service provider without the preceding steps, see Section 5.1.6, below.

5.1.4. Profile Description

The following sections provide detailed definitions of the individual profile steps. If the profile is initiated by the service provider, start with Section 5.1.4.1. If initiated by the identity provider, start with Section 5.1.4.5. In the descriptions below, the following are referred to:

Single Sign-On Service

This is the authentication request protocol endpoint at the identity provider to which the <AuthnRequest> message is delivered by the user agent.

Assertion Consumer Service

This is the authentication request protocol endpoint at the service provider to which the <Response> message is delivered by the user agent.

5.1.4.1. HTTP Request to Service Provider

In this OPTIONAL step, if the first access is to the service provider, an arbitrary request for a resource can initiate the profile. There are no restrictions on the form of the request. The service provider is free to use any means it wishes to associate the subsequent interactions with the original request. Each of the bindings provide a RelayState mechanism that the service provider MAY use to associate the profile exchange with the original request. The service provider SHOULD reveal as little of the request as possible in the RelayState value unless the use of the profile does not require such privacy measures.

5.1.4.2. Service Provider Determines Identity Provider

This step is implementation-dependent. The service provider MAY use
the SAML identity provider discovery profile, described in
[OASIS.saml-core-2.0-os]. The service provider MAY also choose to
redirect the user agent to another service that is able to determine
an appropriate identity provider. In such a case, the service
provider may issue an <AuthnRequest> (as in the next step) to this
service to be relayed to the identity provider, or it may rely on the
intermediary service to issue an <AuthnRequest> message on its
behalf.

5.1.4.3. <AuthnRequest> Is Issued by Service Provider to Identity
Provider

Once an identity provider is selected, the location of its single
sign-on service is determined. Metadata (as in [OASIS.saml-metadata-
2.0-os]) MAY be used for this purpose. In response to an HTTP
request by the user agent, an HTTP response is returned containing an
/AuthnRequest/> message, to be delivered to the identity provider’s
single sign-on service, by the user agent.

The exact format of this HTTP response and the subsequent HTTP
request to the single sign-on service is defined by the SAML binding
used. This profile mandates use of
[OASIS.draft-hodges-saml-binding-noxmldsig-01]. Profile-specific
rules for the contents of the <AuthnRequest> message are included in
Section 5.1.5.1.

HTTP exchanges in this step SHOULD be made over either TLS [RFC4346]
or SSL [SSL3] to maintain confidentiality and message integrity. The
/AuthnRequest/> message SHOULD be signed, per the binding
specification [OASIS.draft-hodges-saml-binding-noxmldsig-01].

The identity provider MUST process the <AuthnRequest> message as
described in [OASIS.saml-core-2.0-os]. This may constrain the
subsequent interactions with the user agent, for example if the
IsPassive attribute is utilized.

5.1.4.4. Identity Provider Identifies Principal

At any time during the previous step or subsequent to it, the
identity provider MUST establish the identity of the principal
(unless it returns an error to the service provider). The ForceAuthn
/AuthnRequest/> attribute, if present with a value of "true", obligates
the identity provider to freshly establish this identity, rather than
relying on an existing session it may have with the principal.
Otherwise, and in all other respects, the identity provider may use
any means to authenticate the user agent, subject to any requirements
included in the <AuthnRequest> in the form of the
<RequestedAuthnContext/> element.
5.1.4.5. Identity Provider Issues <Response> to Service Provider

Regardless of the success or failure of the <AuthnRequest>, the identity provider SHOULD produce an HTTP response to the user agent containing a <Response> message, depending on the SAML binding used, to be delivered to the service provider's assertion consumer service.

The exact format of this HTTP response and the subsequent HTTP request to the assertion consumer service is defined by the SAML binding used. Profile-specific rules on the contents of the <Response> are included in Section 5.1.5.2. Since the HTTP POST binding is used in this profile, the <Response> message is delivered directly to the service provider in this step.

The location of the assertion consumer service MAY be determined using metadata (as in [OASIS.saml-metadata-2.0-os]). The identity provider MAY employ some out-of-scope means to establish that this location is in fact controlled by the service provider. A service provider MAY indicate the SAML binding and the specific assertion consumer service to use in its <AuthnRequest> and the identity provider SHOULD honor them if it can.

The HTTP requests in this step SHOULD be made over TLS [RFC4346] to maintain confidentiality and message integrity. The <Response> message SHOULD be signed per the binding specification [OASIS.draft-hodges-saml-binding-noxmldsig-01].

The service provider MUST process the <Response> message and any enclosed <Assertion> elements as described in [OASIS.saml-core-2.0-os].

5.1.4.6. Service Provider Grants or Denies Access to User Agent

To complete the profile, the service provider processes the <Response> and <Assertion>(s) and grants or denies access to the resource. The service provider MAY establish a security context with the user agent using any session mechanism it chooses, e.g. using [RFC2965]. Any subsequent use of the <Assertion>(s) provided are at the discretion of the service provider and other relying parties, subject to any restrictions on use contained within said assertions.

5.1.5. Use of Authentication Request Protocol

This profile is based on the Authentication Request protocol defined in [OASIS.saml-core-2.0-os]. In the nomenclature of actors enumerated in Section 3.4 of that document, the service provider is the request issuer and the relying party, and the principal is the presenter, requested subject, and confirming entity. There may be
additional relying parties or confirming entities at the discretion of the identity provider (see below).

5.1.5.1. <AuthnReq> Usage

A service provider MAY include any message content described in [OASIS.saml-core-2.0-os], Section 3.4.1. All processing rules are as defined in [OASIS.saml-core-2.0-os]. The <Issuer> element MAY be present and if so, it SHOULD contain the unique identifier of the requesting service provider; the Format attribute MUST be omitted or have a value of:

urn:oasis:names:tc:SAML:2.0:nameid-format:entity

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

The identity provider MUST include a <NameIDPolicy> element with the AllowCreate XML attribute set to "true".

The service provider MUST NOT include <Subject> or <Conditions> elements in the request.

Note that if the <AuthnRequest> is not authenticated and/or integrity protected -- i.e. it is not signed -- the parties are taking on certain risks. These are discussed more fully in the Security Considerations section below.

5.1.5.2. <Response> Usage and Assertion Profile

If the identity provider wishes to return an error, it MUST NOT include an assertion in the <Response> message. Otherwise, if the request is successful -- or if the response is not associated with a request (see Unsolicited Response section) -- the <Response> element, and the <Assertion> it contains, MUST conform to the following:

The <Issuer> element MAY be omitted, but if present it MUST contain the unique identifier of the issuing identity provider; the Format attribute MUST be omitted or have a value of:

urn:oasis:names:tc:SAML:2.0:nameid-format:entity

It MUST contain exactly one <Assertion> element. The assertion’s <Issuer> element MUST contain an identifier denoting the issuing identity provider; the Format attribute MUST be omitted or have a value of:
urn:oasis:names:tc:SAML:2.0:nameid-format:entity

The assertion MUST contain at least one <AuthnStatement> that reflects the authentication of the principal to the identity provider.

The assertion MUST contain an <AuthnStatement>, which itself MUST contain a <Subject> element with at least one <SubjectConfirmation> element containing a Method of:

urn:oasis:names:tc:SAML:2.0:cm:bearer

The SessionIndex XML attribute MUST NOT be included.

The bearer <SubjectConfirmation> element described above MUST contain a <SubjectConfirmationData> element that contains a Recipient XML attribute containing the service provider’s assertion consumer service URL and a NotOnOrAfter XML attribute that limits the window during which the assertion can be delivered. It MAY contain an Address XML attribute limiting the client address from which the assertion can be delivered. It MUST NOT contain a NotBefore XML attribute. If the containing message is in response to an <AuthnRequest>, then the InResponseTo XML attribute MUST match the request’s ID.

Other statements and confirmation methods MAY be included in the assertion at the discretion of the identity provider. In particular, <AttributeStatement> elements MAY be included. The <AuthnRequest> MAY contain an AttributeConsumingServiceIndex XML attribute referencing information about desired or required attributes in [OASIS.saml-metadata-2.0-os]. The identity provider MAY ignore this, or send other attributes at its discretion.

The assertion MUST contain an <AudienceRestriction> identifying the service provider in the <Audience>.

Other conditions (and other <Audience> elements) MAY be included as requested by the service provider or at the discretion of the identity provider. Of course, all such conditions SHOULD be understood by and accepted by the service provider in order for the assertion to be considered valid. Though doing so is at the service provider’s discretion. The identity provider is NOT obligated to honor the requested set of <Conditions> in the <AuthnRequest>, if any.
5.1.5.3. <Response> Message Processing Rules

Regardless of the SAML binding used, the service provider MUST do the following:

- Verify any signatures present on the assertion(s) or the response.
- Verify that the Recipient attribute in any bearer <SubjectConfirmationData> matches the assertion consumer service URL to which the <Response> was delivered.
- Verify that the NotOnOrAfter XML attribute in any bearer <SubjectConfirmationData> has not passed, subject to allowable clock skew between the providers.
- Verify that the InResponseTo XML attribute in the bearer <SubjectConfirmationData> equals the ID of its original <AuthnRequest> message, unless the response is unsolicited (see Section 5.1.6), in which case the InResponseTo XML attribute MUST NOT be present.
- Verify that any assertions relied upon are valid in other respects.

If any bearer <SubjectConfirmationData> includes an Address XML attribute, the service provider MAY check the user agent’s client address against it.

Any assertion which is not valid, or whose subject confirmation requirements cannot be met MUST be discarded and MUST NOT be used to establish a security context for the principal.

If an <AuthnStatement> used to establish a security context for the principal contains a SessionNotOnOrAfter XML attribute, the security context SHOULD be discarded once this time is reached, unless the service provider reestablishes the principal’s identity by repeating the use of this profile.

5.1.5.4. POST-specific Processing Rules

The service provider MUST ensure that bearer assertions are not replayed, by maintaining the set of used ID values for the length of time for which the assertion would be considered valid based on the NotOnOrAfter XML attribute in the <SubjectConfirmationData>.

5.1.6. Unsolicited Responses

An identity provider MAY initiate this profile by delivering an
unsolicited <Response> message to a service provider.

An unsolicited <Response> MUST NOT contain an InResponseTo XML attribute, nor should any bearer <SubjectConfirmationData> elements contain one. If metadata as specified in [OASIS.saml-metadata-2.0-os] is used, the <Response> SHOULD be delivered to the <md:AssertionConsumerService> endpoint of the service provider designated as the default.

Of special mention is that the identity provider MAY include a binding-specific "RelayState" parameter that indicates, based on mutual agreement with the service provider, how to handle subsequent interactions with the user agent. This MAY be the URL of a resource at the service provider. The service provider SHOULD be prepared to handle unsolicited responses by designating a default location to send the user agent subsequent to processing a response successfully.

5.1.7. Use of Metadata

Please refer to [OASIS.saml-profiles-2.0-os] section 4.1.6 for metadata usage guidelines. See also [OASIS.saml-metadata-2.0-os].

5.2. Example SAML Assertion

This section presents an example SAML assertion.

In the first example, Figure 2, the assertion is attesting with respect to the subject (lines 7-15) "Alice@example.com" (line 11). The validity conditions are expressed in lines 16-23, via both a validity period expressed as temporal endpoints, and an "audience restriction" stating that this assertion’s semantics are valid for only the relying party named "example2.com". Also, the assertion’s issuer is noted in lines 4-5. The authentication statement in lines 24-30 indicate that the issuer is attesting to having authenticated the subject using the mechanism denoted in the <AuthnContext> element.
5.3. Security Considerations

This section discusses security considerations this profile’s security considerations. The reader should also refer to [OASIS.saml-sec-consider-2.0-os].

5.3.1. Unintended Principal Data Leakage

This profile does not require the identity provider to validate the service provider’s <AssertionConsumerServiceURL> or <AssertionConsumerServiceInde> as stipulated in the section 4.1.4.1 of the SAMLv2 Web Browser SSO profile specified in [OASIS.saml-
Additionally, the Lightweight SSO profile specified herein does not require service providers to sign their <AuthnRequest> messages, and thus an identity provider does not, in that case, have cryptographically-based assurance as to whom it is responding to.

Both of the above items, either together or alone, serve to facilitate arbitrary runtime interactions between identity providers and service providers, however the Principal is vulnerable in these situations to unintended leakage of personal identity information ("PII") to unintended, and perhaps malevolent, parties.

### 5.3.2. Man-in-the-middle Attacks and Stolen Assertions

**Threat:**

Stolen assertion via a MITM

The attacker could then impersonate the subject (the putative caller) at the service provider.

**Countermeasures:**

SAML’s classic approach to this is to have the identity provider sign the assertions. When assertions are integrity-protected and there is data origination authentication, SAML assertion’s various content attesting to the issuer, subject, audience, and validity periods serve to reduce risk of a service provider relying upon a replayed assertion.

In this profile however, signing is optional. If the SP accepts unsigned <Response> messages, then it is leaving itself explicitly open to "Man In The Middle" attacks, with all the attendant downsides.

### 5.3.3. Forged Assertion

**Threat:**

A malicious user or user agent could forge or alter a SAML assertion in order to communicate with the service provider since the user agent is used as a conduit.

**Countermeasures:**

To avoid this kind of attack, the entities must assure that proper mechanisms for protecting the SAML assertion are employed, e.g.,
signing the SAML <Response> message.

5.4. Contributors
@@ TODO.

5.5. Acknowledgments
@@ TODO.

5.6. IANA Considerations

This document contains a number of IANA considerations. A future version of this document will list them in this section.

5.7. Open Issues

None at this time.
6. References

6.1. Normative References

[OASIS.draft-hodges-saml-binding-noxmldsig-01]

[OASIS.saml-bindings-2.0-os]

[OASIS.saml-core-2.0-os]

[OASIS.saml-metadata-2.0-os]

[OASIS.saml-profiles-2.0-os]

[OASIS.saml-sec-consider-2.0-os]


6.2. Informative References

[IANA.application.samlassertion-xml]
OASIS Security Services Technical Committee (SSTC),
"application/samlassertion+xml MIME Media Type Registration", IANA MIME Media Types Registry application/samlassertion+xml, December 2004.

[OASIS.saml-conformance-2.0-os]

[OASIS.saml-glossary-2.0-os]

[OASIS.sstc-saml-exec-overview-2.0-cd-01]

[OASIS.sstc-saml-protocol-ext-thirdparty-cd-01]

[OASIS.sstc-saml-tech-overview-2.0-draft-08]


[W3C.xmldsig-core]
Eastlake, D., Reagle, J., and D. Solo, "XML-Signature
Authors’ Addresses

Jeff Hodges
NeuStar, Inc.
2000 Broadway Street
Redwood City, CA  94063
US
Email: Jeff.Hodges@neustar.biz

Scott Cantor
Internet2
B320-17 KRC-BLDG B -- OSU
Columbus, OH  43212
US
Email: cantor.2@osu.edu
Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.