

SIP
Internet-Draft
Intended status: Experimental
Expires: September 9, 2009

H. Tschofenig
Nokia Siemens Networks
J. Hodges
Unaffiliated
J. Peterson
NeuStar, Inc.
J. Polk
Cisco
D. Sicker
CU Boulder
March 8, 2009

SIP SAML Profile and Binding
draft-ietf-sip-saml-06.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on September 9, 2009.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights

Internet-Draft

SIP SAML

March 2009

and restrictions with respect to this document.

Internet-Draft

SIP SAML

March 2009

Abstract

This document specifies a Session Initiation Protocol (SIP) profile of Security Assertion Markup Language (SAML) as well as a SAML SIP binding. The defined SIP SAML Profile composes with the mechanisms defined in the SIP Identity specification and satisfy requirements presented in "Trait-based Authorization Requirements for the Session Initiation Protocol (SIP)".

Internet-Draft

SIP SAML

March 2009

Table of Contents

1.	Introduction	5
2.	Terminology	6
3.	SAML Introduction	8
3.1.	SAML Assertions	9
3.2.	Abstract Request/Response Protocol	10
4.	Specification Scope	11
5.	Employing SAML in SIP	12
6.	SIP SAML Profiles	14
6.1.	AS-driven SIP SAML URI-based Attribute Assertion Fetch Profile	14
6.1.1.	Required Information	14
6.1.2.	Profile Overview	14
6.1.3.	Profile Description	18
6.1.4.	Assertion Profile Description	21
6.1.5.	Assertion Verification	23
6.2.	Caller-driven SIP SAML Conveyed Assertion Profile	24
7.	SAML SIP Binding	26
7.1.	SAML HTTP-URI-based SIP Binding	26
8.	Example SAML Assertions	27
9.	Authentication Service Behavior	32
10.	Verifier Behavior	34
11.	SAML-Info Header Field	36
12.	Extended RFC 4474 SIP Identity Signature Mechanism	38
13.	Security Considerations	41
13.1.	Man-in-the-Middle Attacks and Stolen Assertions	41
13.2.	Forged Assertion	41
13.3.	Replay Attack	42

14.	Contributors	43
15.	Acknowledgments	44
16.	IANA Considerations	45
16.1.	Header Field Names	45
16.2.	477 'Binding to SIP Message failed' Response Code	45
16.3.	478 'Unknown SAML Assertion Content' Response Code	45
16.4.	479 'Invalid SAML Assertion' Response Code	46
16.5.	480 'Use SAML Header' Response Code	46
17.	Change Log	47
17.1.	-05 to -06	47
17.2.	-04 to -05	47
17.3.	-03 to -04	47
17.4.	-02 to -03	47
17.5.	-00 to -02	48
18.	References	49
18.1.	Normative References	49
18.2.	Informative References	50
	Authors' Addresses	52

[1.](#) Introduction

This document specifies composition of the Security Assertion Markup Language (SAML) V2.0 with SIP [[RFC3261](#)] in order to accommodate richer authorization mechanisms and enable "trait-based authorization." Trait-based authorization is where one is authorized to make use of some resource based on roles or traits rather than ones identifier(s). Motivations for trait-based authorization, along with use-case scenarios, are presented in [[RFC4484](#)].

Security Assertion Markup Language (SAML) v2.0, "SAMLv2", is an XML-based framework for creating and exchanging security information. [[OASIS.sstc-saml-exec-overview-2.0-cd-01](#)] and [[OASIS.sstc-saml-tech-overview-2.0-draft-16](#)] provide non-normative overviews of SAMLv2. The SAMLv2 specification set is normatively defined by [[OASIS.saml-conformance-2.0-os](#)].

Various means of providing trait-based authorization exist: authorization certificates [[RFC3281](#)], SPKI [[RFC2693](#)], or extensions to the authenticated identity body [[RFC3893](#)]. The authors selected SAML due to its increasing use in environments, such as the Liberty Alliance, and the Internet2 project, areas where the applicability to

SIP is widely desired.

[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](#)].

The SIP network element "Authentication Service" is introduced in [[RFC4474](#)]. We reuse this term to refer to a network element that authenticates and authorizes a user and creates a "SIP identity assertion". This system entity is the logical equivalent of a "SAML Authority" in the SAML terminology.

For overall SIP terminology, see [[RFC3261](#)].

In this specification, the term, or term component, "SAML" refers to

SAML V2.0 in all cases. For example, the term "SAML assertion" implicitly means "SAMLv2 assertion". For overall SAML terminology, see [[OASIS.saml-glossary-2.0-os](http://www.oasis-open.org/committees/document.php?doc_id=3133)].

The below list maps other various SIP terms to their SAML (rough-)equivalents:

Element, Network Element:

System Entity, Entity

Authentication Service:

SAML Authority

Invitee, Invited User, Called Party, Callee:

Relying Party

Server, User Agent Server (UAS):

SAML Responder

User Agent Client (UAC), client:

SAML Requester

Additional terms defined in the context of this specification:

profile attribute(s):

one or more attributes of a "user profile".

user profile, subject profile:

the set of various attributes accompanying (i.e., mapped to) a user account in many environments.

SAML [[OASIS.sstc-saml-exec-overview-2.0-cd-01](#)] [[OASIS.sstc-saml-tech-overview-2.0-draft-16](#)] 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 an assertion itself, or the subject of an assertion.

Thus, one can employ SAML to make and 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 conveyed in, or over, another protocol is known as a "SAML Binding". Typically, a SAML profile specifies the SAML bindings that may be used in its context. Both SAML profiles and SAML bindings reference other 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:

- o 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.
- o 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:

[3.1.](#) 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.

[3.2.](#) 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](#)]). Trait-based SIP communication session establishment, the topic of this specification, is another.

[4.](#) Specification Scope

The scope of this specification is:

- o Specify a SIP profile of SAML -- also known as a "SIP SAML profile" -- such that a subject's profile attributes. In doing so, satisfy the requirements outlined in [[RFC4484](#)].

The following are outside the scope of this specification:

- o Defining a means for configuring the runtime behavior, or deployment characteristics, of the Authentication Service.

Discussion:

For example, a SIP Authentication Service could be implemented such that its SAML-based features are employed, or not, on a subject-by-subject basis, and/or on a domain-by-domain basis.

- o The definition of specific conveyed subject profile attributes (aka traits).

Discussion:

This specification defines a facility enabling "trait-based authorization" as discussed in [[RFC4484](#)].

The attributes of interest in trait-based authorization will be ones akin to, for example: roles, organizational membership, access rights, or authentication event context. Definition of such attributes is application- and/or deployment-context-dependent and are not defined in this specification. However, The

SAMLv2 specification defines several "SAML Attribute Profiles" for encoding attributes from various application domains, e.g., LDAP, UUID/GUID, DCE PAC, and XACML, in SAML assertions [[OASIS.saml-profiles-2.0-os](#)].

In order for any trait-based system to be practical, participating entities must agree on attributes and traits that will be conveyed and subsequently relied upon. Without such agreements, a trait-based system cannot be usefully deployed. This specification does not discuss the manner in which participating entities might discover one another or agree on the syntax and semantics of attributes and traits.

Note that SAMLv2 specifies a "metadata" facility that may be useful in addressing this need.

[5.](#) Employing SAML in SIP

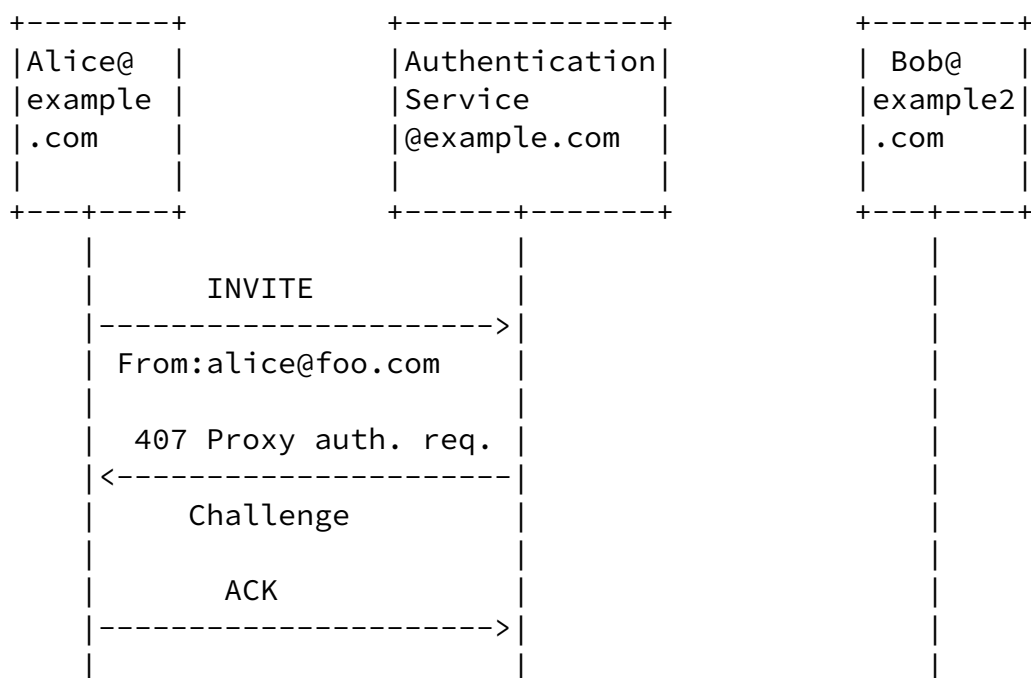
Employing SAML in SIP necessitates devising a new SAML profile(s) and binding(s) because those already specified in the SAMLv2 specification set are specific to other use contexts, e.g., HTTP-based web browsing. Although SIP bears some similarity to HTTP, it is a separately distinct protocol, thus requiring specification of SIP-specific SAML profile(s) and binding(s).

The SIP SAML Profiles defined in this document make use of concepts defined by [[RFC4474](#)] "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)" -- also known as "SIP Identity". In particular, they leverage the "mediated authentication architecture" utilizing the Authentication Service (AS). The overall semantic being that the AS is vouching that it did indeed authenticate the calling party.

Such an Authentication Service, which likely has access to various pieces of information concerning the calling party, could also act as a SAML Authority, and make such information available to the callee via SAML.

The approach used by this document is similar to the one used for SIP Identity, i.e. the AS creates a SAML assertion and makes it available to the verifier via a reference, in the particular case of the AS-

driven SIP SAML URI-based Attribute Assertion Fetch Profile. Figure 1 illustrates this approach in a high-level summary fashion. Figure 2, further below, illustrates additional details. In case of the Assertion-by Value profile the SAML assertion is made available to the verifying party directly without the additional step of utilizing a reference. This approach is described in [Section 6.2](#).



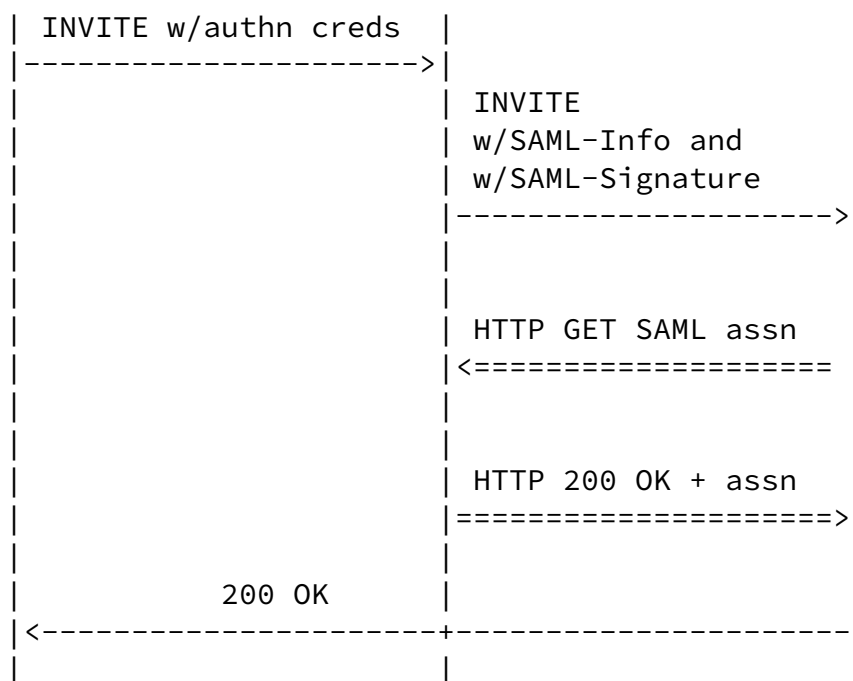


Figure 1: SIP-SAML-based Network Asserted Identity

Figure 1 shows an exchange based on the AS-driven SIP SAML URI-based Attribute Assertion Fetch Profile where the AS creates a SAML assertion, creates a reference to it, and puts that reference into the SAML-Info header before forwarding the SIP message. To tie the SAML-Info field to the message a digital signature is computed and placed in the SAML-Signature header. Bob in our case acting as the verifier uses the reference to retrieve the SAML assertion, verifies it and the SAML-Signature.

[6.](#) SIP SAML Profiles

This section defines two "SIP SAML profiles":

- o The "AS-driven SIP SAML URI-based Attribute Assertion Fetch Profile"
- o The "Assertion-by-value" Profile

[6.1.](#) AS-driven SIP SAML URI-based Attribute Assertion Fetch Profile

[6.1.1.](#) 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:sip:sip-saml-profile:as:uri:attr:1.0

Contact Information:

Hannes Tschofenig (Hannes.Tschofenig@nsn.com)

SAML Confirmation Method Identifiers:

The SAML V2.0 confirmation method identifier is used in this profile.

Description:

Given below.

Updates:

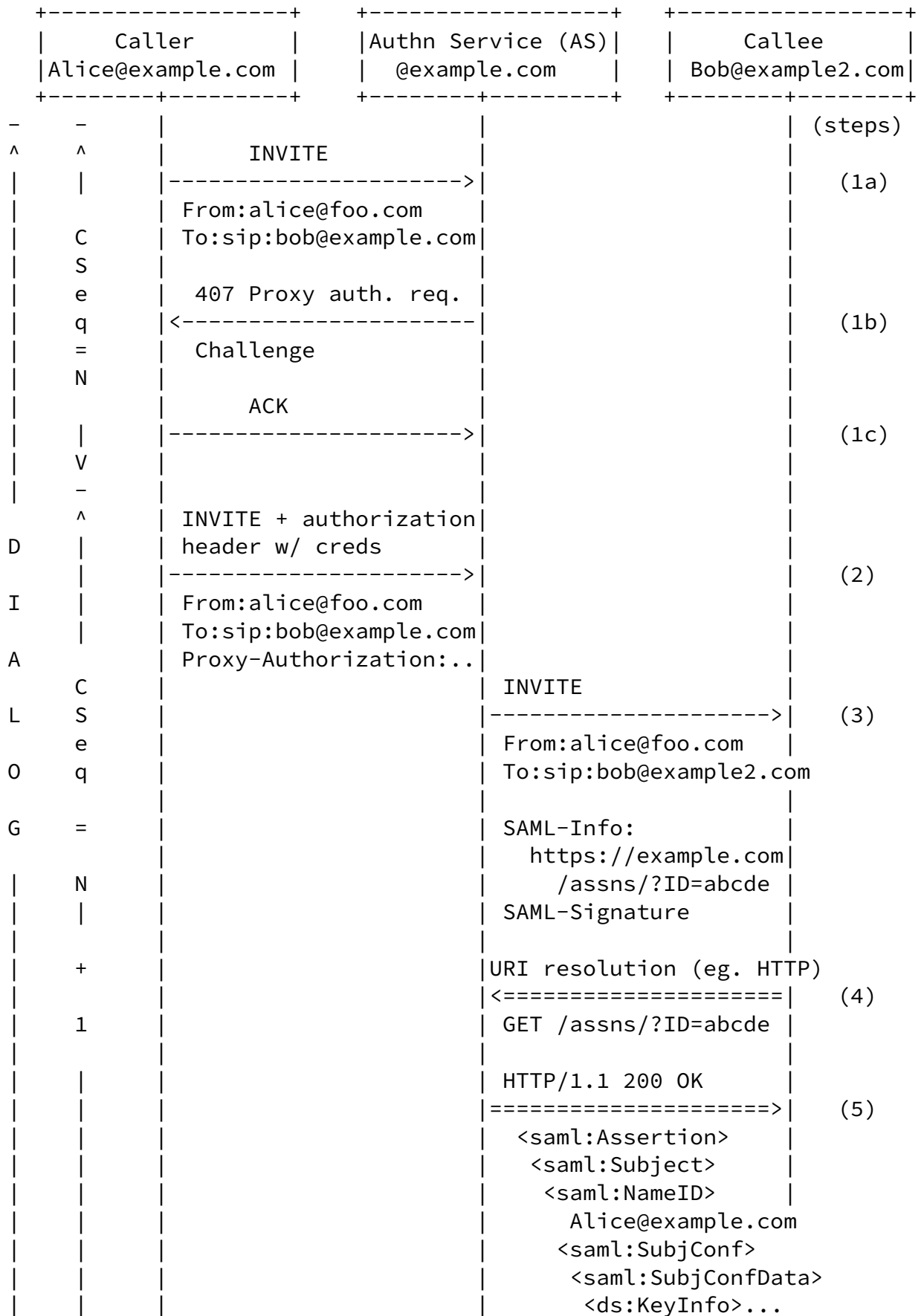
None.

[6.1.2.](#) Profile Overview

Figure 2 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.

Although this profile is overview is cast in terms of a SIP INVITE transaction, the reader should note that the mechanism specified herein, may be applied to any SIP request message.

Figure 2 begins on the next page.



Internet-Draft

SIP SAML

March 2009

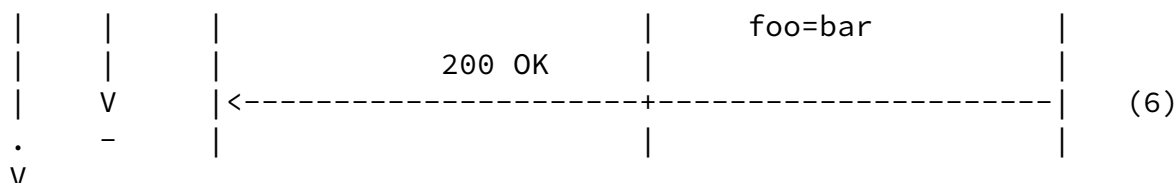


Figure 2: AS-driven SIP SAML Attribute Fetch Profile: Example INVITE Transaction

Step 1. Initial SIP Transaction between Caller and AS

This optional initial step is comprised of substeps 1a, 1b, and 1c in Figure 2. In this step, the caller, Alice, sends a SIP request message, illustrated as an INVITE, indicating Bob as the callee (1a), is subsequently challenged by the AS (1b), and sends an ACK in response to the challenge (1c). The latter message signals the completion of this SIP transaction (which is an optional substep of this profile).

Step 2. Caller sends SIP Request Message with Authorization Credentials to the AS.

Alice then sends an INVITE message in response to the challenge, or uses cached credentials for the domain if step 1 was skipped, as specified in [RFC4474] and [RFC3261]. Depending on the chosen SIP security mechanism for client authentication either digest authentication, client side authentication of Transport Layer Security, or a combination of both is used to provide the AS with a strong assurance about the identity of Alice.

Step 3. AS authorizes the SIP Request and Forwards it to Callee.

First, the AS authorizes the received INVITE message, as specified in [RFC4474] and [RFC3261]. If the authorization procedure is successful, the AS creates a SAML assertion asserting Alice's profile attributes required by Bob's domain (example2.com)., and also containing a the domain's (example.com) public key certificate, or a reference to it.

The AS constructs a HTTP-based SAML URI Reference incorporating the assertion's Assertion ID (see [Section 2.3.3](#) of [[OASIS.saml-core-2.0-os](#)]). The AS uses this URI as the value for the SAML-Info header it adds to the INVITE message.

The AS determines which profile attributes (if any) to assert in the <AttributeStatement> via local configuration

and/or obtaining example2.com's metadata [[OASIS.saml-metadata-2.0-os](#)]. The AS then sends the updated INVITE message to Bob.

Step 4. Callee Dereferences HTTP-based SAML URI Reference

Bob's UAC or SIP Proxy receives the message and needs to obtain Alice's domain certificate that is contained in the SAML assertion. It obtains the HTTP-based SAML URI Reference from the message's SAML-Info header and dereferences it per [Section 7.1](#). Note that this is not a SIP message, but an HTTP message [[RFC2616](#)].

Step 5. AS Returns SAML Assertion

Upon receipt of the above HTTP request, which contains an embedded reference to Alice's SAML Assertion, Alice's AS returns her assertion in an HTTP response message.

Upon receipt of Alice's SAML Assertion, the binding between the SAML assertion and the SIP message is verified. A detailed description can be found in [Section 10](#). Various elements contained in the SAML assertion are inspected and the processing of the INVITE message is continued.

Step 6. Callee Returns SIP 200 OK to Caller

If Bob determines, based upon Alice's identity as asserted by the AS, and as further substantiated by the information in the SAML assertion, to accept the INVITE, he returns a SIP 200 OK message directly to Alice.

[6.1.3](#). Profile Description

The following sections provide detailed definitions of the individual profile steps. The relevant illustration is Figure 3, below. Note that this profile is agnostic to the specific SIP request, and also that the Sender and Authentication Service (AS) may be separate or co-located in actuality.

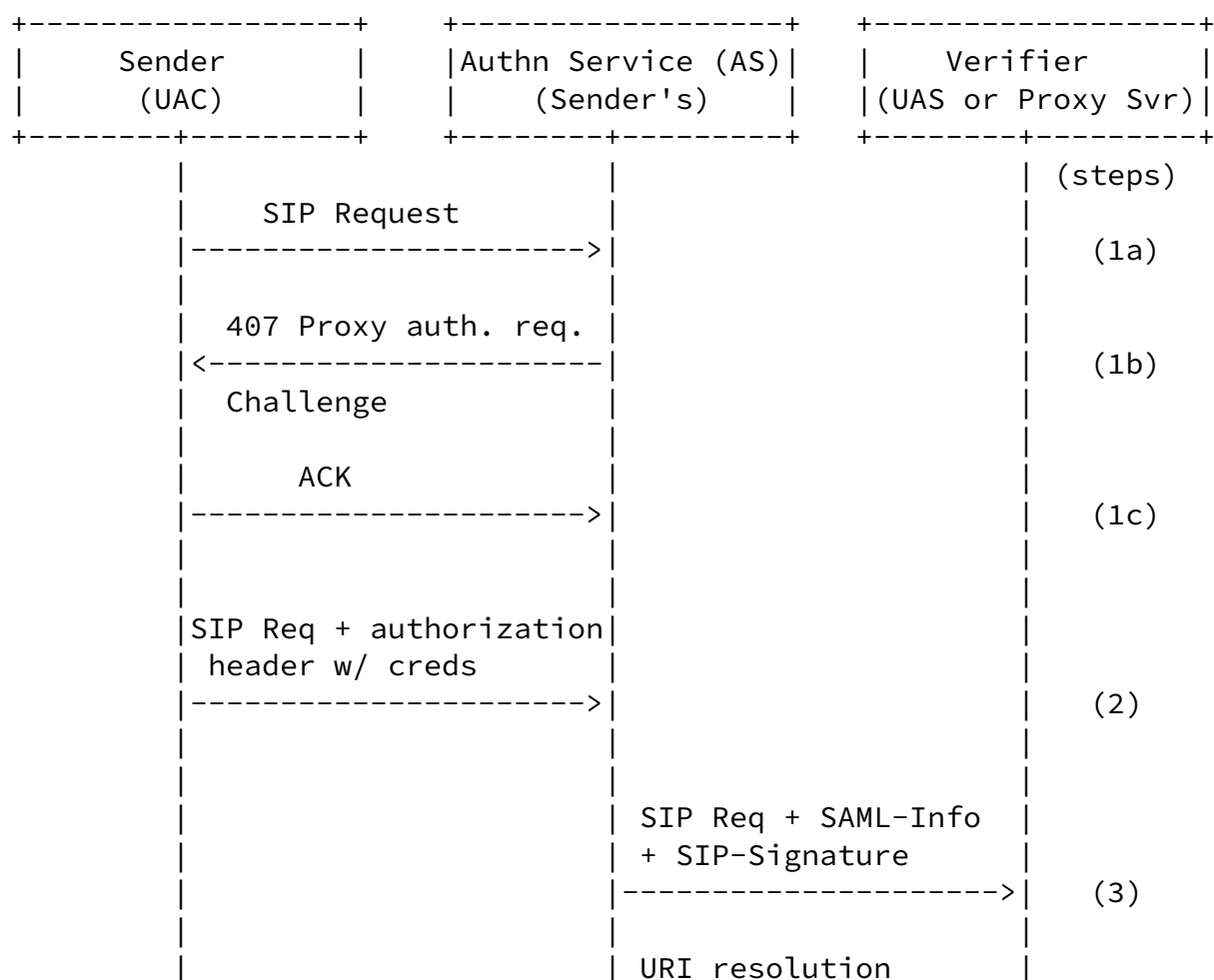




Figure 3: AS-driven SIP SAML Attribute Fetch Profile: Message Flow

[6.1.3.1.](#) Initial SIP Transaction between Sender and AS

This optional step maps to Steps 1 and 2 of [Section 5](#) "Authentication Service Behavior" of [\[RFC4474\]](#). If the SIP request sent by the caller in substep 1a is deemed insufficiently authenticated by the AS per the rules stipulated by [\[RFC4474\]](#) Steps 1 and 2, then the AS MUST authenticate the sender of the message. The particulars of how this is accomplished depend upon implementation and/or deployment instantiation as discussed in [\[RFC4474\]](#). Substeps 1b and 1c as shown

in Figure 3 are non-normative and illustrative only.

[6.1.3.2.](#) Sender sends SIP Request Message with Authorization Credentials to the AS

This step maps to Steps 1 and 2 of [Section 5](#) "Authentication Service Behavior" of [\[RFC4474\]](#). This request is presumed to be made in a context such that the AS will not challenge it -- i.e., the AS will consider the sender of the message to be authenticated. If this is not true, then this procedure reverts back to Step 1, above.

Otherwise, the AS carries out all other processing of the message as stipulated in [\[RFC4474\]](#) Steps 1 and 2, and if successful, this procedure proceeds to the next step below.

[6.1.3.3.](#) AS Authorizes the SIP Request and Forwards it to Verifier

This first portion of this step maps to Steps 3 and 4 of [Section 9](#), which the AS MUST perform, although with the following additional

substeps:

The AS MUST construct a SAML assertion according to the "Assertion Profile Description" specified in [Section 6.1.4](#) of this specification.

The AS MUST construct an HTTP URI per Section "3.7.5.1 URI Syntax" of [\[OASIS.saml-bindings-2.0-os\]](#). To enable proper caching, the HTTP URI pointing to the SAML assertion MUST be unique, i.e., if the content of the SAML assertion changes then the HTTP URI reference MUST be different than any previously used HTTP URI references used before.

The AS MUST use the URI constructed in the immediately preceding substep as the value of the SAML-Info header that is added to the SIP request message.

Upon successful completion of all of the above, the AS forwards the request message.

At this point in this step, after perhaps traversing some number of intermediaries, the SIP request message arrives at a SIP network entity performing the "verifier" role. This role and its behavior are specified in [Section 10](#).

[6.1.3.4](#). Verifier Dereferences HTTP-based SAML URI Reference

The verifier SHOULD ascertain whether it has a current cached copy of the SIP message sender's SAML assertion and domain certificate. If

not, or if the verifier chooses to (e.g., due to local policy), it MUST dereference the the HTTP-based SAML URI Reference found in the SIP message's SAML-Info header. To do so, the verifier MUST employ the "SAML HTTP-URI-based SIP Binding" specified in [Section 7.1](#).

[6.1.3.5](#). AS Returns SAML Assertion

This step also employs [Section 7.1](#) "SAML HTTP-URI-based SIP Binding".

If the prior step returns an HTTP error (e.g., 4xx series), then this procedure terminates and the verifier returns (upstream) a SIP 436 'Bad SAML-Info' Response code.

Otherwise, the HTTP response message will contain a SAML assertion and be denoted as such via the MIME media type of "application/samlassertion+xml" [[IANA.application.samlassertion+xml](#)]. The verifier MUST perform the verification steps specified in [Section 6.1.5](#) "Assertion Verification", below. If successful, then this procedure continues with the next step.

[6.1.3.6](#). Verifier performs Next Step

The SIP request was successfully processed. The verifier now performs its next step, which depends at least in part on the type of SIP request it received.

[6.1.4](#). Assertion Profile Description

This section defines the particulars of how the sender, i.e., the SAML Authority, MUST construct certain portions of the SAML assertions it issues. The schema for SAML assertions themselves is defined in Section 2.3 of [[OASIS.saml-core-2.0-os](#)].

An example SAML assertion, formulated according to this profile is given in [Section 8](#).

In the following subsections, the SAML assertion profile is specified element-by-element, in a top-down, depth-first manner, beginning with the outermost element, "<Assertion>". Where applicable, the requirements for an element's XML attributes are also stated, as a part of the element's description. Requirements for any given element or XML attribute are only stated when, in the context of use of this profile, they are not already sufficiently defined by [[OASIS.saml-core-2.0-os](#)].

[6.1.4.1](#). Element: <Assertion>

Attribute: ID

The value for the ID XML attribute SHOULD be allocated randomly

such that the value meets the randomness requirements specified in Section 1.3.4 of [[OASIS.saml-core-2.0-os](#)].

Attribute: IssueInstant

The value for the IssueInstant XML attribute SHOULD be set at the time the SAML assertion is created (and cached for subsequent retrieval). This time instant value MAY be temporally the same as that encoded in the SIP message's Date header, and MUST be at least temporally later, although it is RECOMMENDED that it not be 10 minutes or more later.

[6.1.4.1.1](#). Element: <Issuer>

The value for the Issuer XML element MUST be a value that matches either the Issuer or the Issuer Alternative Name fields [[RFC3280](#)] in the certificate conveyed by the SAML assertion in the ds:X509Certificate element located on this path within the SAML assertion:

```
<Assertion
  <ds:Signature
    <ds:KeyInfo
      <ds:X509Data
        <ds:X509Certificate
```

This field contains the domain certificate of the AS.

[6.1.4.1.2](#). Element: <Subject>

The <Subject> element SHOULD contain both a <NameID> element and a <SubjectConfirmation> element.

The value of the <NameID> element MUST be the Address of Record (AoR).

The <SubjectConfirmation> element attribute Method SHOULD be set to the value:

urn:oasis:names:tc:SAML:2.0:cm:sender-vouches

Although it MAY be set to some other implementation- and/or deployment-specific value. The <SubjectConfirmation> element itself

SHOULD be empty.

[6.1.4.1.3.](#) Element: <Conditions>

The <Conditions> element SHOULD contain an <AudienceRestriction> element, which itself SHOULD contain an <Audience> element. When included the value of the <Audience> element MUST be the same as the addr-spec of the SIP request's To header field.

The following XML attributes of the <Conditions> element MUST be set as follows:

Attribute: NotBefore

The value of the NotBefore XML attribute MUST be set to a time instant the same as the value for the IssueInstant XML attribute discussed above, or to a later time.

Attribute: NotOnOrAfter

The value of the NotOnOrAfter XML attribute MUST be set to a time instant later than the value for NotBefore.

[6.1.4.1.4.](#) Element: <AttributeStatement>

The SAML assertion MAY contain an <AttributeStatement> element. If so, the <AttributeStatement> element will contain attribute-value pairs, e.g., of a user profile nature, encoded according to either one of the "SAML Attribute Profiles" as specified in [\[OASIS.saml-profiles-2.0-os\]](#), or encoded in some implementation- and/or deployment-specific attribute profile.

The attribute-value pairs SHOULD in fact pertain to the entity identified in the SIP From header field, since a SAML assertion formulated per this overall section is stating that they do.

[6.1.5.](#) Assertion Verification

This section specifies the steps that a verifier participating in this profile MUST perform in addition to the "Verifier Behavior" specified in [Section 6 of \[RFC4474\]](#).

The steps are:

1. Before Step 1 in [Section 6 of \[RFC4474\]](#), the verifier MUST extract the AS's domain certificate from the <ds:X509Certificate> XML element at the end of the element path given in [Section 6.1.4.1.1.](#)

Internet-Draft

SIP SAML

March 2009

2. Perform Step 1 in [Section 6 of \[RFC4474\]](#).
3. After Step 1 in [Section 6 of \[RFC4474\]](#), but before Step 2 of that section, the verifier MUST verify the SAML assertion's signature via the procedures specified in Section 5.4 of [\[OASIS.saml-core-2.0-os\]](#) as well as [\[W3C.xmlsig-core\]](#).

The 479 'Invalid SAML Assertion' response code is used when the verifier is unable to process the SAML assertion.

4. Perform Step 2 in [Section 6 of \[RFC4474\]](#).
5. Verify that the signer of the SIP message's Identity header field is the same as the signer of the SAML assertion.
6. Verify that the content of the SAML assertion, if present, matches with the information carried in the SIP message. This may include the following checks:
- 7.

- * Verify that the SAML assertion's <Issuer> element value matches the Issuer or the Issuer Alternative Name fields [\[RFC3280\]](#) in the AS's domain certificate.
- * Verify that the SAML assertion's <NameID> element value is the same as the Address of Record (AoR) value.
- * Verify that the SAML assertion's <SubjectConfirmation> element value is set to whichever value was configured at implementation- or deployment-time. The default value is:

urn:oasis:names:tc:SAML:2.0:cm:sender-vouches

- * Verify that the SAML assertion contains an <Audience> element, and that its value matches the value of the addr-spec of the SIP To header field.
- * Verify that the validity period denoted by the NotBefore and NotOnOrAfter attributes of the <Conditions> element meets the requirements given in [Section 6.1.4.1.3](#).

[6.2.](#) Caller-driven SIP SAML Conveyed Assertion Profile

For the "Assertion-by-value" profile we assume that a SAML assertion is obtained out-of-band and attached to the body of the SIP message. Note that any SIP message may be used to convey the SAML assertion even though SIP INVITE may be the most appropriate candidate. The

verification step described in [Section 6.1.5](#) is applicable to this profile as well as the description on the content of the assertion illustrated in [Section 6.1.4](#).

[7.](#) SAML SIP Binding

This section specifies one SAML SIP Binding at this time. Additional bindings may be specified in future revisions of this specification. The description in [Section 6.1.4](#) is applicable to this profile.

[7.1.](#) SAML HTTP-URI-based SIP Binding

This section specifies the "SAML HTTP-URI-based SIP Binding", (SHUSB).

The SHUSB is a profile of the "SAML URI Binding" specified in [Section 3.7](#) of [[OASIS.saml-bindings-2.0-os](#)]. The SAML URI Binding specifies a means by which SAML assertions can be referenced by URIs and thus be obtained through resolution of such URIs.

This profile of the SAML URI Binding is congruent with the SAML URI Binding -- including support for HTTP-based URIs being mandatory to implement -- except for the following further restrictions which are specified in the interest of interoperability (section numbers refer to [[OASIS.saml-bindings-2.0-os](#)]):

[Section 3.7.5.3](#) Security Considerations:

Support for TLS 1.0 or SSL 3.0 is mandatory to implement.

[Section 3.7.5.4](#) Error Reporting:

All SHOULDs in this section are to be interpreted as MUSTs.

[8.](#) Example SAML Assertions

This section presents two examples of a SAML assertion, one unsigned (for clarity), the other signed (for accuracy).

In the first example, Figure 4, 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 above items correspond to some aspects of this specification's SAML assertion profile, as noted below in Security Considerations discussions, see: [Section 13.1](#) and [Section 13.2](#).

In lines 24-36, Alice's telephone number is conveyed, in a "typed" fashion, using LDAP/X.500 schema as the typing means.

```
1 <Assertion ID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
2   IssueInstant="2003-04-17T00:46:02Z" Version="2.0"
3   xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
4   <Issuer>
5     example.com
6   </Issuer>
7   <Subject>
8     <NameID
9       Format=
10        "urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
11        Alice@example.com
12      </NameID>
13    <SubjectConfirmation
14      Method="urn:oasis:names:tc:SAML:2.0:cm:sender-vouches"/>
```

```

15    </Subject>
16    <Conditions NotBefore="2003-04-17T00:46:02Z"
17        NotOnOrAfter="2003-04-17T00:51:02Z">
18        <AudienceRestriction>
19            <Audience>
20                example2.com
21            </Audience>
22        </AudienceRestriction>
23    </Conditions>
24    <AttributeStatement>
25        <saml:Attribute
26            xmlns:x500=
27                "urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
28            NameFormat=
29                "urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
30            Name="urn:oid:2.5.4.20"
31            FriendlyName="telephoneNumber">
32            <saml:AttributeValue xsi:type="xs:string">
33                +1-888-555-1212
34            </saml:AttributeValue>
35        </saml:Attribute>
36    </AttributeStatement>
37 </Assertion>

```

Figure 4: Unsigned SAML Assertion Illustrating Conveyance of Subject Attribute

In the second example, Figure 5, the information described above is the same, the addition is that this version of the assertion is signed. All the signature information is conveyed in the <ds:signature> element, lines 7-47. Thus this assertion's origin and its integrity are assured. Since this assertion is the same as the one in the first example above, other than having a signature added, the

second example below addresses the same Security Considerations aspects, plus those requiring a Signature.


```

1 <Assertion ID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
2   IssueInstant="2003-04-17T00:46:02Z" Version="2.0"
3   xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
4   <Issuer>
5     example.com
6   </Issuer>
7   <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
8     <ds:SignedInfo>
9       <ds:CanonicalizationMethod
10         Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
11       <ds:SignatureMethod
12         Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
13       <ds:Reference
14         URI="#_a75adf55-01d7-40cc-929f-dbd8372ebdfc">
15         <ds:Transforms>
16           <ds:Transform
17             Algorithm=
18             "http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
19           <ds:Transform
20             Algorithm=
21             "http://www.w3.org/2001/10/xml-exc-c14n#">
22             <InclusiveNamespaces
23               PrefixList="#default saml ds xs xsi"
24               xmlns=
25               "http://www.w3.org/2001/10/xml-exc-c14n#" />
26             </ds:Transform>
27           </ds:Transforms>
28           <ds:DigestMethod
29             Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
30           <ds:DigestValue>
31             Kclet6Xca0g0WXM4gty6/UNdviI=
32           </ds:DigestValue>
33         </ds:Reference>
34       </ds:SignedInfo>
35       <ds:SignatureValue>
36         hq4zk+ZknjggCQgZm7ea8fI7...Hr7wHxvCCRwubfZ6RqVL+wNmeWI4=
37       </ds:SignatureValue>
38       <ds:KeyInfo>
39         <ds:X509Data>
40           <ds:X509Certificate>
41             MIIcYjCCAj0gAwIBAgICAnUwDQYJKoZIhvcNAQEEBQAwgaxNVBAYTA1VT
42             MRIwEAYDVQQIEw1XaXNjb ..... dnP6Hr7wHxvCCRwubnZAv2FU78pLX
43             8I3bsbmRAUg4UP9hH6ABVq4KQKMknxu1xQxLhpR1ylGPdioG8cCx3w/w==
44           </ds:X509Certificate>
45         </ds:X509Data>
46       </ds:KeyInfo>
47     </ds:Signature>
48   <Subject>

```

Internet-Draft

SIP SAML

March 2009

```
49     <NameID
50       Format=
51       "urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
52       Alice@example.com
53     </NameID>
54     <SubjectConfirmation
55       Method="urn:oasis:names:tc:SAML:2.0:cm:sender-vouches"/>
56   </Subject>
57   <Conditions NotBefore="2003-04-17T00:46:02Z"
58     NotOnOrAfter="2003-04-17T00:51:02Z">
59     <AudienceRestriction>
60       <Audience>
61         example2.com
62       </Audience>
63     </AudienceRestriction>
64   </Conditions>
65   <AttributeStatement>
66     <saml:Attribute
67   xmlns:x500=
68     "urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
69   NameFormat=
70     "urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
71   Name="urn:oid:2.5.4.20"
72   FriendlyName="telephoneNumber">
73       <saml:AttributeValue xsi:type="xs:string">
74         +1-888-555-1212
75       </saml:AttributeValue>
76     </saml:Attribute>
77   </AttributeStatement>
78 </Assertion>
```

Figure 5: Signed SAML Assertion Illustrating Conveyance of Subject Attribute

[9.](#) Authentication Service Behavior

[RFC4474] defined a new role for SIP entities called an authentication service. This document re-uses the concept and hence the same constraints and properties apply to this document. The subsequent text is copied from [[RFC4474](#)] and modified to fit to the usage of SAML.

Any entity that instantiates the authentication service role MUST possess the private key of a domain certificate. Intermediaries that instantiate this role MUST be capable of authenticating one or more SIP users that can register in that domain. Commonly, this role will be instantiated by a proxy server, since these entities are more likely to have a static hostname, hold a corresponding certificate, and have access to SIP registrar capabilities that allow them to authenticate users in their domain. It is also possible that the authentication service role might be instantiated by an entity that acts as a redirect server, but that is left as a topic for future work.

SIP entities that act as an authentication service MUST add a Date header field to SIP requests if one is not already present. Similarly, authentication services MUST add a Content- Length header field to SIP requests if one is not already present; this can help verifiers to double-check that they are hashing exactly as many bytes of message-body as the authentication service when they verify the message.

Entities instantiating the authentication service role perform the following steps, in order, to generate an Identity header for a SIP request:

Step 1:

The authentication service MUST extract the identity of the sender from the request. The authentication service takes this value

from the From header field; this AoR will be referred to here as the 'identity field'. If the identity field contains a SIP or SIP Secure (SIPS) URI, the authentication service MUST extract the hostname portion of the identity field and compare it to the domain(s) for which it is responsible (following the procedures in [RFC 3261, Section 16.4](#), used by a proxy server to determine the domain(s) for which it is responsible). If the identity field uses the TEL URI scheme, the policy of the authentication service determines whether or not it is responsible for this identity. If the authentication service is not responsible for the identity in question, it SHOULD process and forward the request normally, but it MUST NOT add a SAML-Info and a SAML-Signature header.

Step 2:

The authentication service MUST determine whether or not the sender of the request is authorized to claim the identity given in the identity field. In order to do so, the authentication service MUST authenticate the sender of the message.

Step 3:

The authentication service SHOULD ensure that any preexisting Date header in the request is accurate. Local policy can dictate precisely how accurate the Date must be; a RECOMMENDED maximum discrepancy of ten minutes will ensure that the request is unlikely to upset any verifiers. If the Date header contains a time different by more than ten minutes from the current time noted by the authentication service, the authentication service SHOULD reject the request. This behavior is not mandatory because a user agent client (UAC) could only exploit the Date header in order to cause a request to fail verification; the SAML-Signature header is not intended to provide a source of non-repudiation or a perfect record of when messages are processed. Finally, the authentication service MUST verify that the Date header falls within the validity period of its certificate.

Step 4:

The authentication service MUST form the signature and add the

SAML-Signature header to the request containing this signature. After the SAML-Signature header has been added to the request, the authentication service MUST also add an SAML-Info header. The SAML-Info header contains a URI from which the SAML assertion and a domain certificate can be acquired.

Finally, the authentication service MUST forward the message normally.

[10](#). Verifier Behavior

When a verifier receives a SIP message containing an SAML-Signature and SAML-Info header, it may inspect these two header fields. Typically, the results of a verification are provided as input to an authorization process that is outside the scope of this document. If an SAML-Info and SAML-Signature header is not present in a request, and one is required by local policy (for example, based on a per-sending-domain policy, or a per-sending-user policy), then a 428 'Use SAML Header' response MUST be sent.

In order to verify the identity of the sender of a message, an entity acting as a verifier MUST perform the following steps, in the order here specified.

Step 1:

The verifier has to acquire the certificate for the signing domain. Implementations supporting this specification SHOULD have some means of retaining domain certificates (in accordance with normal practices for certificate lifetimes and revocation) in order to prevent themselves from needlessly downloading the same

certificate every time a request from the same domain is received. Certificates cached in this manner should be indexed by the URI given in the SAML-Info header field value.

Provided that the domain certificate used to sign this message is not previously known to the verifier, SIP entities SHOULD discover this certificate by dereferencing the SAML-Info header, unless they have some more efficient implementation-specific way of acquiring certificates for that domain. The domain certificate can be found in the SAML assertion, either by reference or by value. The verifier processes this certificate in the usual ways, including checking that it has not expired, that the chain is valid back to a trusted certification authority (CA), and that it does not appear on revocation lists. Once the certificate is acquired, it MUST be validated following the procedures in [RFC 3280](#) [RFC3280]. If the certificate cannot be validated (it is self-signed and untrusted, or signed by an untrusted or unknown certificate authority, expired, or revoked), the verifier MUST send a 437 'Unsupported Certificate' response.

Step 2:

The verifier MUST follow the process described in [Section 13.4 of RFC4474](#) to determine if the signer is authoritative for the URI in the From header field.

Step 3:

The verifier MUST verify the signature in the SAML-Signature header field, following the procedures for generating the hashed digest-string described in [Section 12](#). If a verifier determines that the signature on the message does not correspond to the reconstructed digest-string, then a 479 'Invalid SAML Assertion' response MUST be returned.

Step 4:

The verifier MUST validate the Date, Contact, and Call-ID headers. It must furthermore ensure that the value of the Date header falls within the validity period of the certificate whose corresponding

private key was used to sign the Identity header.

11. SAML-Info Header Field

This document introduces the SIP header field "SAML-Info" to carry a reference to a SAML assertion. This header MAY appear in any SIP header and MAY also appear more than once.

The grammar for this header is (following the ABNF [[RFC4234](#)] in [Section 25 of RFC 3261](#) [[RFC3261](#)]):


```

SAML-Info      =
    "SAML-Info" HCOLON ident-info *( SEMI ident-info-params )

ident-info      = LAQUOT absoluteURI RAQUOT

ident-info-params = generic-param

```

Figure 6: SAML-Info ABNF grammar

The "absoluteURI" portion of the SAML-Info header MUST contain a URI which dereferences to a resource containing a SAML assertion. All implementations of this specification MUST support the use of HTTP and HTTPS URIs in the SAML-Info header. Such HTTP and HTTPS URIs MUST follow the conventions of [RFC 2585](#) [[RFC2585](#)], and for those URIs the indicated resource MUST be of the form 'application/samlassertion+xml' described in that specification.

No parameters are defined for the SAML-Info header in this document. Future experimental RFCS may define additional SAML-Info header parameters.

This document adds the following entries to Table 2 of [RFC 3261](#) [[RFC3261](#)]:

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
-----	-----	-----	---	---	---	---	---	---
SAML-Info	R	a	o	o	-	o	o	o
			SUB	NOT	REF	INF	UPD	PRA
			---	---	---	---	---	---
			o	o	o	o	o	o

Figure 7: New SAML-Info Row for [RFC3261](#) Table 2

The SAML-Info header MUST NOT appear in CANCEL.

12. Extended [RFC 4474](#) SIP Identity Signature Mechanism

To allow the SIP Identity mechanism [[RFC4474](#)] to protect also the reference to the SAML assertion additional SIP header fields need to be protected by the signature calculation mechanisms. The extended signature computation is included in the SAML-Signature header field.

The grammar for this new header is (following the ABNF [[RFC4234](#)] in [RFC 3261](#) [[RFC3261](#)]):

```
SAML-Signature = "SAML-Signature" HCOLON ( signed-identity-digest
                                     sig-info-fields sig-info-alg ) / sig-info-extension
signed-identity-digest = LDQUOTE 32LHEX RDQUOTE
sig-info-alg = "alg" EQUAL token
sig-info-fields = "fields" EQUAL token
sig-info-extension = generic-param
```

The signed-identity-digest is a signed hash of a canonical string generated from certain components of a SIP request. To create the contents of the signed-identity-digest, the following elements of a SIP message MUST be placed in a bit-exact string in the order specified here, separated by a vertical line, "|" or %x7C, character:

- o The AoR of the UA sending the message, or addr-spec of the From header field (referred to occasionally here as the 'identity field').
- o The addr-spec component of the To header field, which is the AoR to which the request is being sent.
- o The callid from Call-Id header field.
- o The digit (1*DIGIT) and method (method) portions from CSeq header field, separated by a single space (ABNF SP, or %x20). Note that the CSeq header field allows linear whitespace (LWS) rather than SP to separate the digit and method portions, and thus the CSeq header field may need to be transformed in order to be canonicalized. The authentication service MUST strip leading zeros from the 'digit' portion of the Cseq before generating the digest-string.
- o The Date header field, with exactly one space each for each SP and the weekday and month items case set as shown in BNF in [RFC 3261](#).

[RFC 3261](#) specifies that the BNF for weekday and month is a choice amongst a set of tokens. The [RFC 2234](#) rules for the BNF specify

that tokens are case sensitive. However, when used to construct the canonical string defined here, the first letter of each week and month MUST be capitalized, and the remaining two letters must be lowercase. This matches the capitalization provided in the definition of each token. All requests that use the Identity mechanism MUST contain a Date header.

- o The addr-spec component of the Contact header field value. If the request does not contain a Contact header, this field MUST be empty (i.e., there will be no whitespace between the fourth and fifth "|" characters in the canonical string).
- o The sig-info-params parameter contains a list of SIP header fields whose values have to be included into the signature calculation. The individual field names in small letters are encoded in the token parameter of the sig-info-fields, each name separated by a "|" character.
- o The body content of the message with the bits exactly as they are in the Message (in the ABNF for SIP, the message-body). This includes all components of multipart message bodies. Note that the message-body does NOT include the CRLF separating the SIP headers from the message-body, but does include everything that follows that CRLF. If the message has no body, then message-body will be empty, and the final "|" will not be followed by any additional characters.

The precise formulation of this digest-string is, therefore (following the ABNF [[RFC4234](#)] in [RFC 3261](#) [[RFC3261](#)]):

```
digest-string = addr-spec "|" addr-spec "|" callid "|"
1*DIGIT SP Method "|" SIP-date "|" [ addr-spec ] "|"
sigfields "|" message-body
```

The sigfields parameter represent the concatenation of the values of the SIP header fields that are included in the signature calculation.

Note again that the first addr-spec MUST be taken from the From header field value, the second addr-spec MUST be taken from the To header field value, and the third addr-spec MUST be taken from the Contact header field value, provided the Contact header is present in the request.

After the digest-string is formed, it MUST be hashed and signed with the certificate for the domain. The hashing and signing algorithm is specified by the 'alg' parameter. This document defines only one

value for the 'alg' parameter: 'rsa-sha1'. All implementations of this specification MUST support 'rsa-sha1'. When the 'rsa-sha1' algorithm is specified in the 'alg' parameter of Identity-Info, the hash and signature MUST be generated as follows: compute the results of signing this string with sha1WithRSAEncryption as described in [RFC 3370](#) [RFC3370] and base64 encode the results as specified in [RFC 3548](#) [RFC3548]. A 1024-bit or longer RSA key MUST be used. The result is placed in the SAML-Signature header field.

This document adds the following entries to Table 2 of [RFC 3261](#) [RFC3261]:

Header field		where	proxy	ACK	BYE	CAN	INV	OPT	REG
-----		----	-----	---	---	---	---	---	---
SAML-Signature		R	a	o	o	-	o	o	o
SUB	NOT	REF	INF	UPD	PRA				
---	---	---	---	---	---				
o	o	o	o	o	o				

Note, in the table above, that this mechanism does not protect the CANCEL method. The CANCEL method cannot be challenged, because it is hop-by-hop, and accordingly authentication service behavior for CANCEL would be significantly limited. Note as well that the REGISTER method uses Contact header fields in very unusual ways that complicate its applicability to this mechanism, and the use of Identity with REGISTER is consequently a subject for future study, although it is left as optional here for forward-compatibility reasons. The SAML-Signature header MUST NOT appear in CANCEL.

[13.](#) Security Considerations

This section discusses security considerations when using SAML with SIP.

[13.1.](#) Man-in-the-Middle Attacks and Stolen Assertions

Threat:

By making SAML assertions available via HTTP-based requests by a potentially unbounded set of requesters, it is conceivably possible that anyone would be able to simply request one and obtain it. By SIP intermediaries on the signaling path for example. Or, an HTTP intermediary/proxy could intercept the assertion as it is being returned to a requester.

The attacker could then conceivably attempt to impersonate the subject (the putative caller) to some SIP-based target entity.

Countermeasures:

Such an attack is implausible for several reasons. The primary reason is that a message constructed by an imposter using a stolen assertion that conveys the public key certificate of some domain will not verify because the values in the SAML assertion, which

are tied to the SIP message, will not verify.

Also, the SIP SAML assertion profile specified herein that the subject's SAML assertion must adhere to causes it to be not useful to arbitrary parties. The subject's assertion:

- * should be signed, thus causing any alterations to break its integrity and make such alterations detectable.
- * relying party is represented in the SAML assertion's Audience Restriction.
- * Issuer is represented in the SAML assertion.
- * validity period for assertion is restricted.

13.2. Forged Assertion

Threat:

A malicious user could forge or alter a SAML assertion in order to communicate with the SIP entities.

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 assertion itself. Section 5.1 of [\[OASIS.saml-core-2.0-os\]](#) specifies the signing of SAML assertions.

Additionally, the assertion content dictated by the SAML assertion profile herein ensures ample evidence for a relying party to verify the assertion and its relationship with the received SIP request.

13.3. Replay Attack

Threat:

Theft of SIP message protected by the mechanisms described herein and replay of it at a later time.

Countermeasures:

The SAML assertion may contain several elements to prevent replay attacks. There is, however, a clear tradeoff between the replaying an assertion and re-using it over multiple SIP exchanges/sessions.

[14.](#) Contributors

The authors would like to thank Marcus Tegnander and Henning Schulzrinne for his contributions to earlier versions of this document.

[15.](#) Acknowledgments

We would like to thank RL 'Bob' Morgan, Stefan Goeman, Shida Schubert, Jason Fischl, Sebastian Felis, Nie Pin, Marcos Dytz, Erkki

Koivusalo, Richard Barnes, Marc Willekens, Marc Willekens, Steffen Fries and Vijay Gurbani for their comments to this draft. The "AS-driven SIP SAML URI-based Attribute Assertion Fetch Profile" is based on an idea by Jon Peterson.

We would also like to thank Eric Rescorla for his expert review.

16. IANA Considerations

16.1. Header Field Names

This document specifies two new SIP header fields: 'SAML-Info' (see [Section 11](#) and 'SAML-Signature' (see [Section 12](#)). IANA is requested to add these two headers to the header sub-registry under <http://www.iana.org/assignments/sip-parameters>.

Header Name: SAML-Info

Compact Form: y

Header Name: SAML-Signature

Compact Form: y

16.2. 477 'Binding to SIP Message failed' Response Code

This document registers a new SIP response code. It is sent when a verifier receives a SAML assertion but the Subject and Condition elements cannot be matched to the content in the SIP message, i.e., the binding between the SIP message and the SAML assertion cannot be accomplished. This response code is defined by the following information, which has been added to the method and response-code sub-registry under <http://www.iana.org/assignments/sip-parameters>.

Response Code Number: 477

Default Reason Phrase: Binding to SIP Message failed

16.3. 478 'Unknown SAML Assertion Content' Response Code

This document registers a new SIP response code. It is used when the verifier is unable to parse the content of the SAML assertion, because, for example, the assertion contains only unknown elements in the SAML assertion, or the SAML assertion XML document is garbled. This response code is defined by the following information, which has been added to the method and response-code sub-registry under <http://www.iana.org/assignments/sip-parameters>.

Response Code Number: 478

Default Reason Phrase: Unknown SAML Assertion Content

16.4. 479 'Invalid SAML Assertion' Response Code

This document registers a new SIP response code. It is used when the verifier is unable to process the SAML assertion. A verifier may be unable to process the SAML assertion in case the assertion is self-signed, or signed by a root certificate authority for whom the verifier does not possess a root certificate. This response code is defined by the following information, which has been added to the method and response-code sub-registry under <http://www.iana.org/assignments/sip-parameters>.

Response Code Number: 479

Default Reason Phrase: Invalid SAML Assertion

16.5. 480 'Use SAML Header' Response Code

This document registers a new SIP response code. It is used when a SAML-Info and SAML-Signature header is not present in a request, and one is required by local policy (for example, based on a per-sending-domain policy, or a per-sending-user policy). This response code is defined by the following information, which has been added to the method and response-code sub-registry under <http://www.iana.org/assignments/sip-parameters>.

Response Code Number: 480

Default Reason Phrase: Use SAML Header

[17.](#) Change Log

RFC Editor - Please remove this section before publication.

[17.1.](#) -05 to -06

In response to the review comments by Eric Rescorla a new signature SIP header field was added.

[17.2.](#) -04 to -05

Changed the document type to experimental

Removed option tag

Added the Caller-driven SIP SAML Conveyed Assertion Profile

Defined a new header (SAML-Info)

Changed the description for usage with this new header

Updated security considerations

Minor editorial cleanups

[17.3.](#) -03 to -04

Updated IANA consideration section.

Added option tag

Updated acknowledgments section

Minor editorial changes to the security considerations section

[17.4.](#) -02 to -03

Denoted that this I-D is intended to update [RFC4474](#) per SIP working group consensus at IETF-69. This is the tact adopted in order to address the impedance mismatch between the nature of the URIs specified as to be placed in the Identity-Info header field, and what is specified in [RFC4474](#) as the allowable value of that header field.

Added placeholder "TBD" section for a to-be-determined "call-by-value" profile, per SIP working group consensus at IETF-69.

Removed use-case appendicies (per recollection of JHodges during IETF-69 discussion as being WG consensus, but such is not noted in

Tschofenig, et al.

Expires September 9, 2009

[Page 47]

Internet-Draft

SIP SAML

March 2009

the minutes).

[17.5.](#) -00 to -02

Will detail in -04 rev.

[18.](#) References

[18.1.](#) Normative References

[OASIS.saml-bindings-2.0-os]

Cantor, S., Hirsch, F., Kemp, J., Philpott, R., and E. Maler, "Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard saml-bindings-2.0-os, March 2005.

[OASIS.saml-core-2.0-os]

Cantor, S., Kemp, J., Philpott, R., and E. Maler, "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard saml-core-2.0-os, March 2005.

[OASIS.saml-metadata-2.0-os]

Cantor, S., Moreh, J., Philpott, R., and E. Maler, "Metadata for the Security Assertion Markup Language

(SAML) V2.0", OASIS Standard saml-metadata-2.0-os, March 2005.

[OASIS.saml-profiles-2.0-os]

Hughes, J., Cantor, S., Hodges, J., Hirsch, F., Mishra, P., Philpott, R., and E. Maler, "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard OASIS.saml-profiles-2.0-os, March 2005.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

[RFC2392] Levinson, E., "Content-ID and Message-ID Uniform Resource Locators", [RFC 2392](#), August 1998.

[RFC2585] Housley, R. and P. Hoffman, "Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP", [RFC 2585](#), May 1999.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2616](#), June 1999.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", [RFC 3261](#), June 2002.

[RFC3280] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet

X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", [RFC 3280](#), April 2002.

[RFC3370] Housley, R., "Cryptographic Message Syntax (CMS) Algorithms", [RFC 3370](#), August 2002.

[RFC3515] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", [RFC 3515](#), April 2003.

[RFC3548] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", [RFC 3548](#), July 2003.

- [RFC3553] Mealling, M., Masinter, L., Hardie, T., and G. Klyne, "An IETF URN Sub-namespace for Registered Protocol Parameters", [BCP 73](#), [RFC 3553](#), June 2003.
- [RFC3893] Peterson, J., "Session Initiation Protocol (SIP) Authenticated Identity Body (AIB) Format", [RFC 3893](#), September 2004.
- [RFC4234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 4234](#), October 2005.
- [RFC4484] Peterson, J., Polk, J., Sicker, D., and H. Tschofenig, "Trait-Based Authorization Requirements for the Session Initiation Protocol (SIP)", [RFC 4484](#), August 2006.
- [W3C.xmlldsig-core] Eastlake, D., Reagle, J., and D. Solo, "XML-Signature Syntax and Processing", W3C Recommendation xmlldsig-core, October 2000, <<http://www.w3.org/TR/xmlldsig-core/>>.

18.2. 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] Mishra, P., Philpott, R., and E. Maler, "Conformance Requirements for the Security Assertion Markup Language (SAML) V2.0", OASIS Standard saml-conformance-2.0-os, March 2005.
- [OASIS.saml-glossary-2.0-os]

Hirsch, F., Philpott, R., and E. Maler, "Security and Privacy Considerations for the OASIS Security Markup Language (SAML) V2.0", OASIS Standard saml-sec-consider-2.0-os, March 2005.

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

Madsen, P. and E. Maler, "SAML V2.0 Executive Overview", OASIS SSTC Committee
Draft sstc-saml-exec-overview-2.0-cd-01, April 2005.

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

Cantor, S., "SAML Protocol Extension for Third-Party Requests", OASIS SSTC Committee Draft sstc-saml-protocol-ext-thirdparty-cd-01, March 2006.

[OASIS.sstc-saml-tech-overview-2.0-draft-16]

Ragouzis, N., Hughes, J., Philpott, R., Maler, E., Madsen, P., and T. Scavo, "Security Assertion Markup Language (SAML) V2.0 Technical Overview", OASIS SSTC Working
Draft sstc-saml-tech-overview-2.0-draft-16, May 2008.

[RFC2543] Handley, M., Schulzrinne, H., Schooler, E., and J. Rosenberg, "SIP: Session Initiation Protocol", [RFC 2543](#), March 1999.

[RFC2693] Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., and T. Ylonen, "SPKI Certificate Theory", [RFC 2693](#), September 1999.

[RFC3281] Farrell, S. and R. Housley, "An Internet Attribute Certificate Profile for Authorization", [RFC 3281](#), April 2002.

[RFC3323] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", [RFC 3323](#), November 2002.

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", [RFC 4474](#), August 2006.

Authors' Addresses

Hannes Tschofenig
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland

Phone: +358 (50) 4871445
Email: Hannes.Tschofenig@gmx.net
URI: <http://www.tschofenig.priv.at>

Jeff Hodges
Unaffiliated

Email: Jeff.Hodges@KingsMountain.com

Jon Peterson
NeuStar, Inc.
1800 Sutter St Suite 570
Concord, CA 94520
US

Email: jon.peterson@neustar.biz

James Polk
Cisco
2200 East President George Bush Turnpike
Richardson, Texas 75082
US

Email: jmpolk@cisco.com

Douglas C. Sicker
University of Colorado at Boulder
ECOT 430
Boulder, CO 80309
US

Email: douglas.sicker@colorado.edu

