SIP
Internet-Draft
Intended status: Standards Track
Expires: April 26, 2007

H. Tschofenig
Siemens Networks GmbH & Co KG
J. Hodges
J. Peterson
NeuStar, Inc.
J. Polk
Cisco
D. Sicker
CU Boulder
October 23, 2006

SIP SAML Profile and Binding
draft-ietf-sip-saml-01.txt

Status of this Memo

Copyright Notice

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)".

Table of Contents

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 [I-D.ietf-sipping-trait-authz].

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-08] 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 [I-D.ietf-sip-identity]. 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].

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.

3. 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 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:

- 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:

- 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:

- Specify a SIP profile of SAML -- aka a "SIP SAML profile" -- such that a subject's profile attributes, and their domain's certificate, can be conveyed to a relying party using SAML. In doing so, satisfy the requirements outlined in [I-D.ietf-sipping-trait-authz], and compose with [I-D.ietf-sip-identity].

The following are outside the scope of this specification:

- 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.

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

  Discussion:

  This specification defines a facility enabling "trait-based authorization" as discussed in [I-D.ietf-sipping-trait-authz].

  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 entites 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 the 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 seperately distinct protocol, thus requiring specification of SIP-specific SAML profile(s) and binding(s). This is technically straightforward as both SAML and SIP are explicitly extensible.

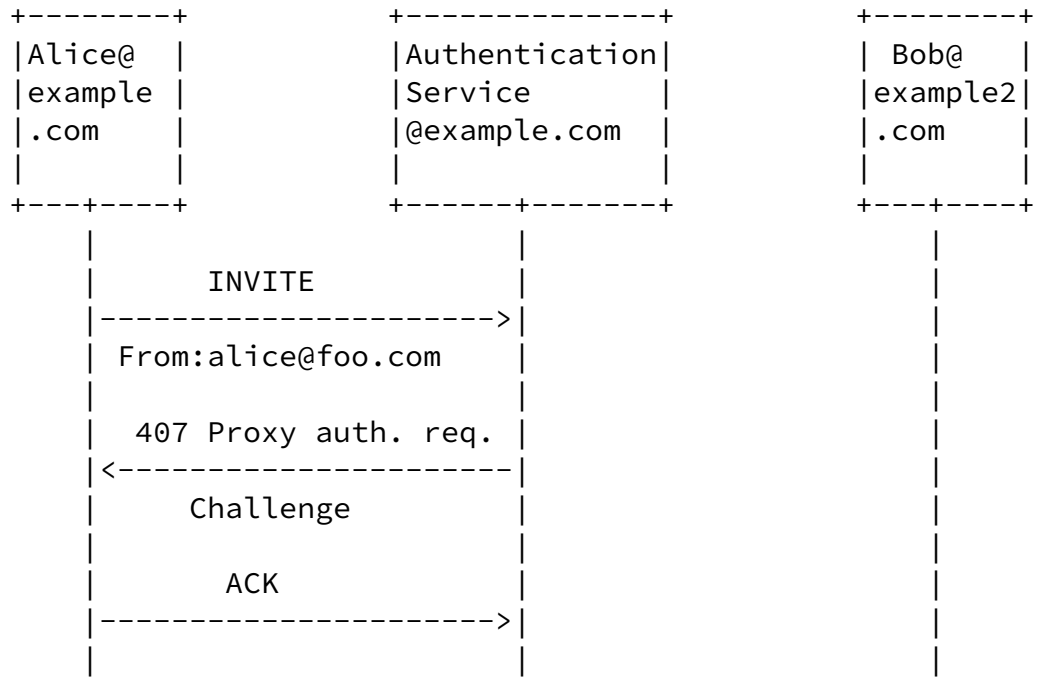
The "Authenticated Identity Management in SIP" specification [I-D.ietf-sip-identity] (aka "SIP Identity") facilitates the composition of SAML and SIP in that it defines a "mediated authentication architecture" where verifying endpoints verify SIP identity assertions -- i.e., the "Identity" header value -- signed by an Authentication Service (AS). The 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.

Since [I-D.ietf-sip-identity] stipulates that the AS must make its certificate available for retrieval and convey the availability and

access mechanism via a URI, in the Identity-Info header, we have an opportunity to compose SIP Identity and SAML.

Such composition can be accomplished by having the resource referred to by the URI in the Identity-Info be a SAML assertion conveying both the AS's certificate and user profile attributes. This is the approach defined in this specification. Figure 1 illustrates this approach in a high-level summary fashion. Figure 2, further below, illustrates additional details.



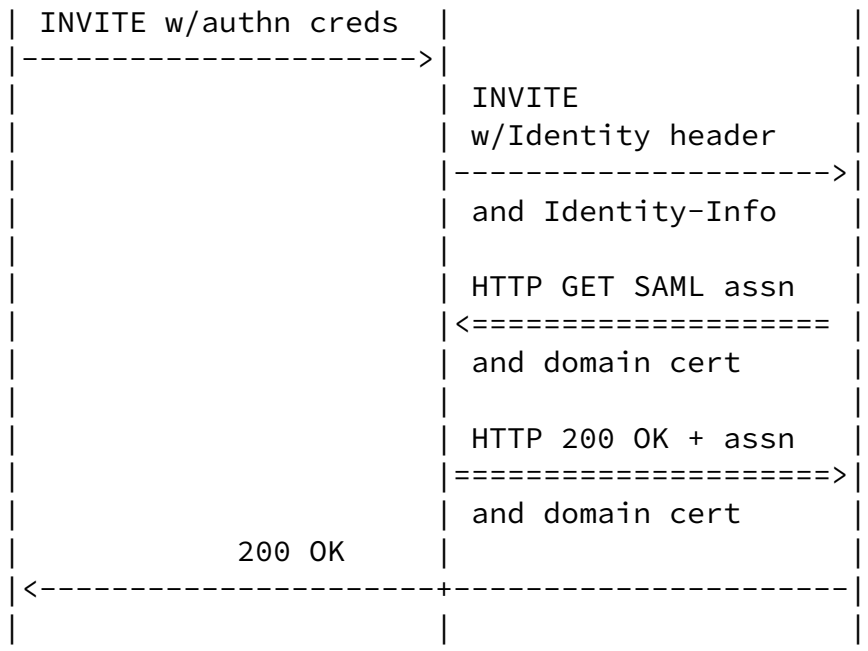


Figure 1: SIP-SAML-based Network Asserted Identity

Since the AS already being trusted to create and add the Identity header containing the SIP Identity Assertion, and to supply a pointer to its domain certificate, having it point instead to a SAML assertion conveying the domain certificate and possibly some user profile attributes, does not significantly alter the first-order security considerations examined in [I-D.ietf-sip-identity]. This specification provides some additional security considerations analysis below in Section 10.

6. SIP SAML Profiles

This section defines one SIP SAML profile:

The "AS-driven SIP SAML URI-based Attribute Assertion Fetch 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

@@ 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:

The SAML V2.0 "{bearer,hok,?}" confirmation method identifier is used in this profile.

Description:

Given below.

Updates:

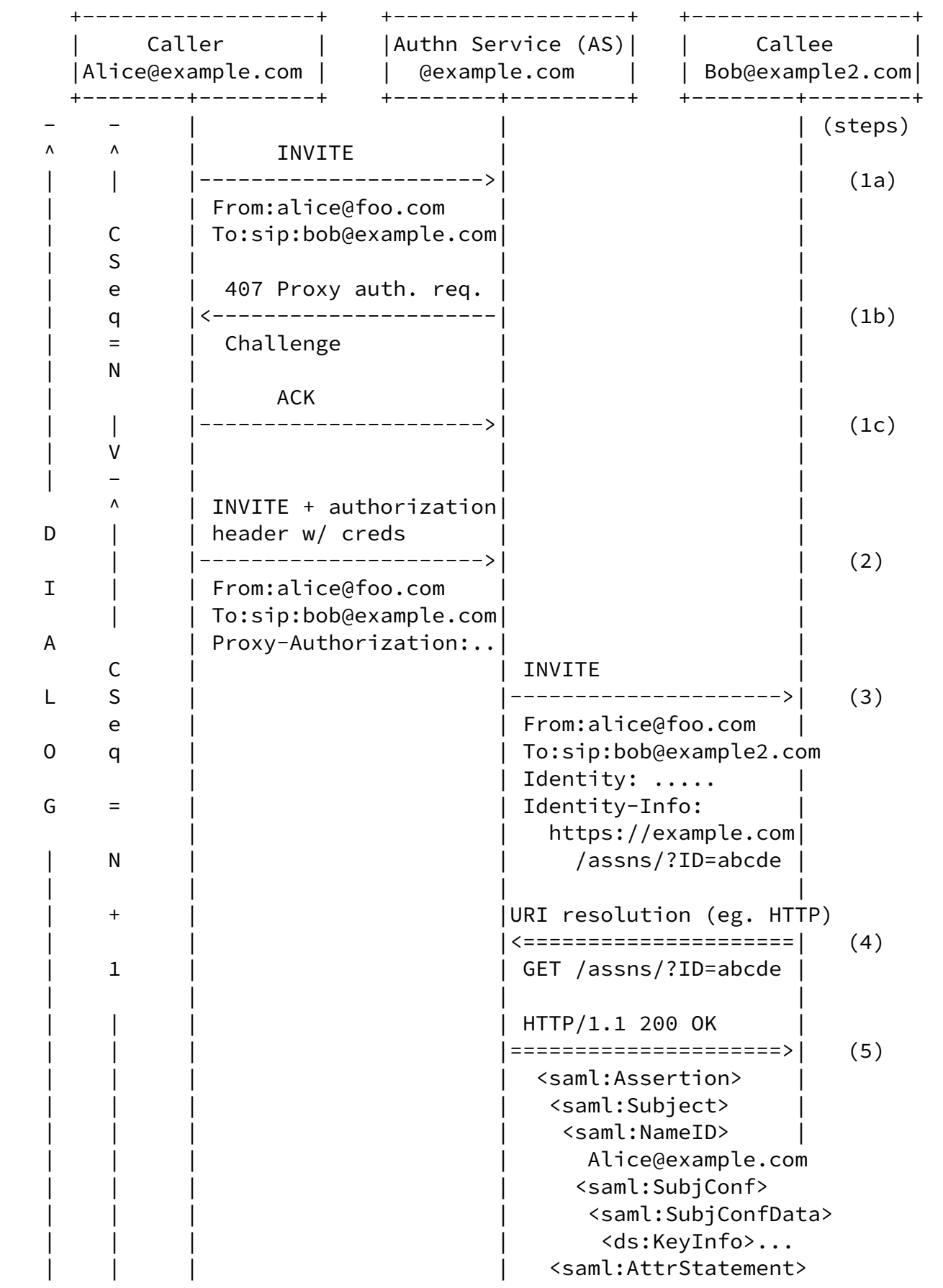
None.

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, and in [I-D.ietf-sip-identity], may be applied to any SIP request message.

Figure 2 begins on the next page.


Caller
Alice@example.com
Authn Service (AS)
@example.com
Callee
Bob@example2.com
(steps)
INVITE
From:alice@foo.com
To:sip:bob@example.com
(1a)
407 Proxy auth. req.
Challenge
(1b)
ACK
(1c)
INVITE + authorization
header w/ creds
From:alice@foo.com
To:sip:bob@example.com
Proxy-Authorization:..
(2)
INVITE
From:alice@foo.com
To:sip:bob@example2.com
Identity: .....
Identity-Info:
https://example.com
/assns/?ID=abcde
(3)
URI resolution (eg. HTTP)
GET /assns/?ID=abcde
(4)
HTTP/1.1 200 OK
(5)
<saml:Assertion>
<saml:Subject>
<saml:NameID>
Alice@example.com
<saml:SubjConf>
<saml:SubjConfData>
<ds:KeyInfo>...
<saml:AttrStatement>
DIALOG
CSeq=N+1

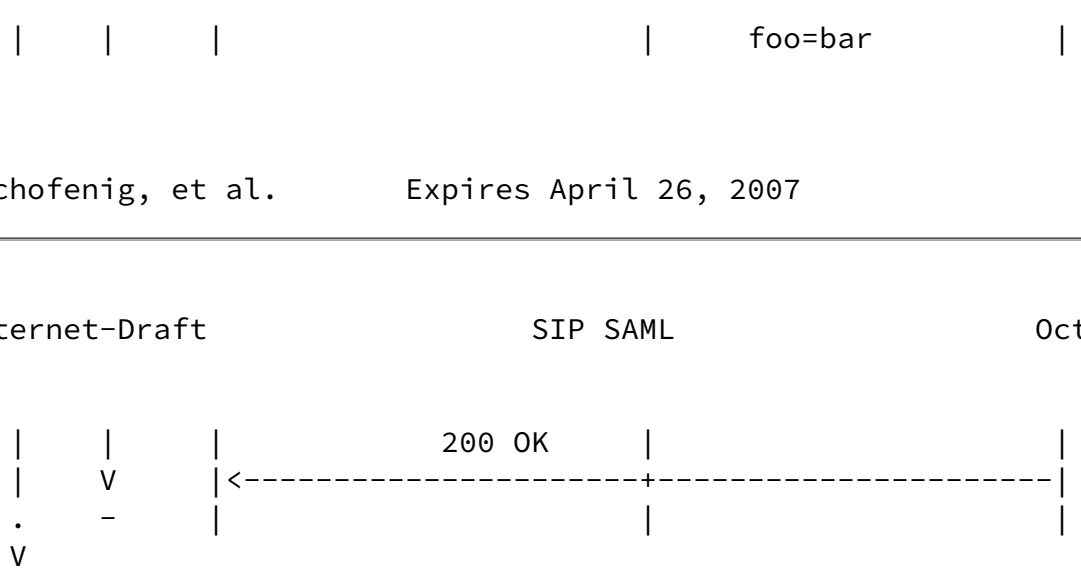


Figure 2: AS-driven SIP SAML Attribute Fetch Profile: Example INVITE Transaction

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 [I-D.ietf-sip-identity] 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 [I-D.ietf-sip-identity] and [RFC3261]. If the authorization is successful, the AS will form the "identity signature" for the message and add Identity and Identity-Info header fields to the message. The AS also at this time constructs and caches 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. This certificate MUST contain the public key corresponding to the private key used to construct the signature whose value was placed in the Identity header. 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 Identity-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 begins verifying it per the "Verifier Behavior" specified in [I-D.ietf-sip-identity]. In order to accomplish this task, it needs to obtain Alice's domain certificate. It obtains the HTTP-based SAML URI Reference from the message's Identity-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 AS continues its verification of the INVITE message. If successful, it returns a 200 OK message directly to Alice. Otherwise it returns an appropriate SIP error response.

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 seperate or co-located in actuality.

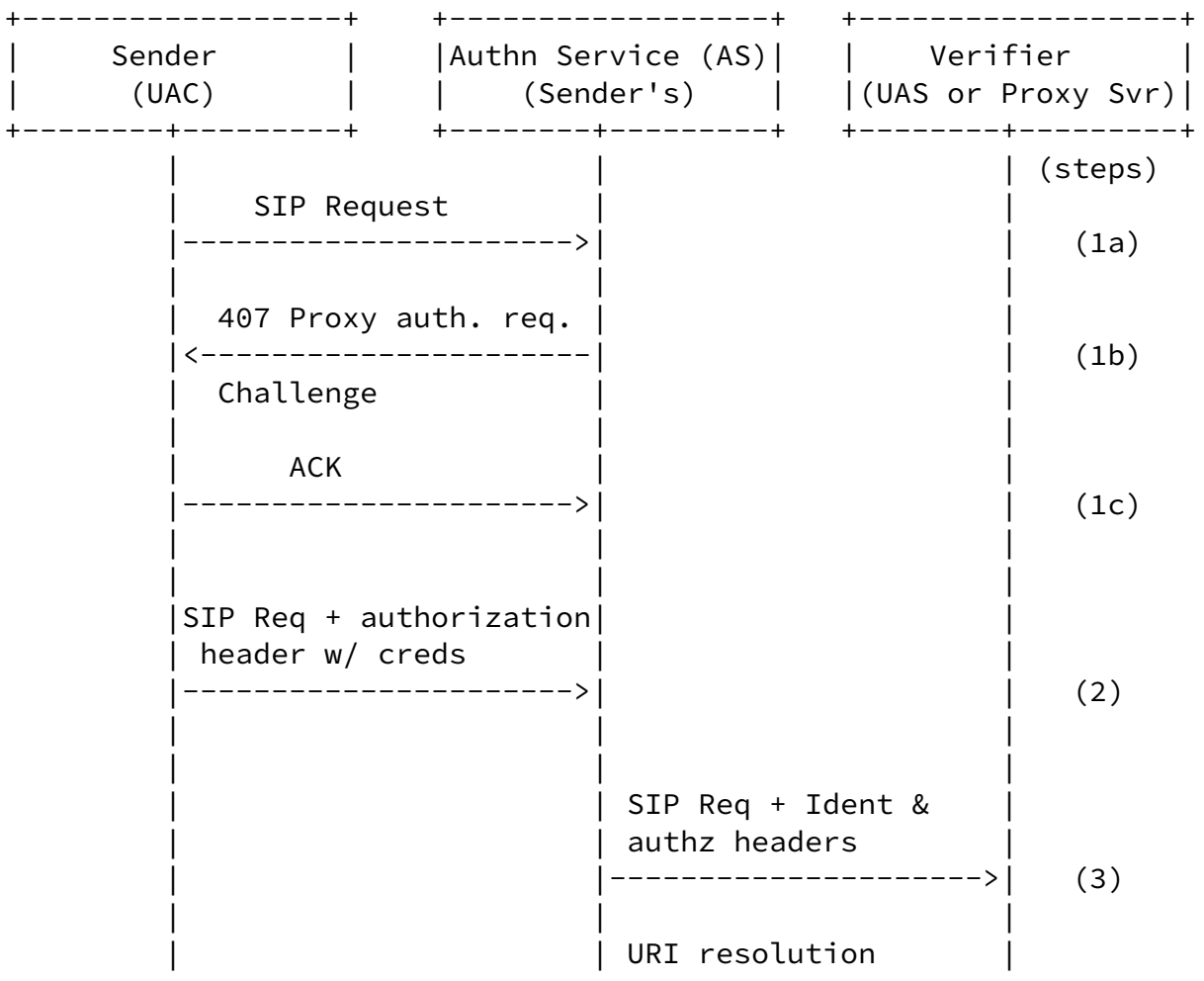


Figure 3: AS-driven SIP SAML Attribute Fetch Profile: Message Flow

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 [I-D.ietf-sip-identity]. If the SIP request sent by the caller in substep 1a is deemed insufficiently authenticated by the AS per the rules stipulated by [I-D.ietf-sip-identity] 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 [I-D.ietf-sip-identity]. 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 [I-D.ietf-sip-identity]. 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 [I-D.ietf-sip-identity] Steps 1 and 2, and if successful, this procedure procedes 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 5

"Authentication Service Behavior" of [I-D.ietf-sip-identity], 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 SHOULD construct an HTTPS, and MAY construct an HTTP, URI per Section "3.7.5.1 URI Syntax" of [OASIS.saml-bindings-2.0-os].
>
> The AS MUST use the URI constructed in the immediately preceding substep as the value of the Identity-Info header that is added to the SIP request message per Step 4 of Section 5 of [I-D.ietf-sip-identity].

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 6 "Verifier Behavior" of [I-D.ietf-sip-identity]. The verifier MUST perform the steps enumerated in the aforementioned section, with the following modifications:

> Step 1 of [I-D.ietf-sip-identity] Section 6 maps to and is updated by, the following two steps in this procedure.

> Steps 2, 3, and 4 of [I-D.ietf-sip-identity] Section 6 may be mapped across this latter portion of this step, and/or the following two steps, as appropriate.

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 Identity-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 Identity-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 9.

Overall SAML assertion profile requirements:

The SAML assertion MUST be signed by the same key as used to sign the contents of the Identity header field. Signing of SAML assertions is defined in Section 5.4 of [OASIS.saml-core-2.0-os].

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 requirments 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
```

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 same as the Address of Record (AoR) value used in computing the "signed-identity-digest" which forms the value of the Identity header. See Section 9 of [I-D.ietf-sip-identity].

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. The value of the <Audience> element SHOULD 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 [I-D.ietf-sip-identity].

The steps are:

1. Before Step 1 in Section 6 of [I-D.ietf-sip-identity], 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.

2. Perform Step 1 in Section 6 of [I-D.ietf-sip-identity].

3. After Step 1 in Section 6 of [I-D.ietf-sip-identity], 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.xmldsig-core].

   @@ TODO: do we need to define a new SIP error response code for when a SAML assn signature is bad? e.g., '4xx Invalid SAML asssertion'.

4. Perform Step 2 in Section 6 of [I-D.ietf-sip-identity].

5. Verify that the signer of the SIP message's Identity header field is the same as the signer of the SAML assertion.

6. Perform Steps 3 and 4 in Section 6 of [I-D.ietf-sip-identity].

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.

8. Verify that the SAML assertion's <NameID> element value is the same as the Address of Record (AoR) value in the "signed-identity-digest". See Section 9 of [I-D.ietf-sip-identity].

9. 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

10. 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.

11. 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.

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.

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. Error Codes

8.1. 425 (Bad SAML Assertion) Response Code

If a UAS or SIP intermediary detects an error in a request message specific to the location information, a new 4XX level error is created here to indicate a problem with the request message. This document creates and IANA registers the new error code:

425 (Bad SAML Assertion)

The 425 (Bad SAML Assertion) response code is a rejection of the content of a SAML assertion within the original SIP Request indicating the SAML assertion was malformed or not satisfactory for the recipient's purpose or could not be dereferenced. No further action by the UAC is required.

The UAC can use means outside the scope of this document to ensure that subsequent requests are going to contain SAML assertions that are acceptable to the UAS. There is no cross-transaction awareness expected by either the UAS or SIP intermediary as a result of this

error message.

More resolution of the error for which the 425 was generated MAY be included in a Reason header [RFC3326]. For these more granular location specific errors, the 'protocol' in the Reason header is 'SAML', defined in Section 3.4. of RFC 3326 [RFC3326] states that the Reason Header normally is not found in a response. This document extends the use of Reason to include its use within a 425 response.

This new error code is IANA registered in Section 9 of this document. An initial set of error codes can be found in Section 8.

8.2. The SAML Reason Protocol

For use with the Reason header, discussed in Section 8.1, this document defines and IANA registers a new Reason Protocol per RFC 3326 [RFC3326].

Protocol Value	Protocol Cause	Reference
SAML	Status	RFCyyyy (i.e., this document)

8.3. Failure Reasons to be Registered

Here is the list and description of each IANA registered location error reason code. If the location generator were to receive one of these indications in a SIP response, it would be in a Reason header. The protocol field of this Reason header would be "SAML", as defined in Section 8.2. Examples of the Reason header are given for each indication below.

8.3.1. SAML Assertion Content Not Supported

"SAML Assertion Content Not Supported" means the SAML content supplied in the request was not processed even though the recipient understood SAML. If the SAML content is understood, but not desired,

a cause=2 or cause=3 response SHOULD be returned.

Cause value: 1

Default text string: SAML Assertion Content Not Supported

An example usage in a SIP Reason header:

Reason: SAML; cause=1; SAML Assertion Content Not Supported

8.3.2. Authentication Statements Desired Instead

"Authentication Statements Desired Instead" means the SAML assertion supplied in the request was understood and supported, but that the recipient, or an application on the recipient demands authentication statements.

Cause value: 2

Default text string: Authentication Statements Desired Instead

An example usage in a SIP Reason header:

Reason: SAML; cause=2; Authentication Statements Desired Instead

8.3.3. Authorization Statements Desired Instead

"Authorization Statements Desired Instead" means the SAML assertion supplied in the request was understood and supported, but that the recipient, or an application on the recipient demands authorization statements.

Cause value: 3

Default text string: Authorization Statements Desired Instead

An example usage in a SIP Reason header:

Reason: SAML; cause=3; Authorization Statements Desired Instead

8.3.4. Attribute Statements Desired Instead

"Attribute Statements Desired Instead" means the SAML assertion supplied in the request was understood and supported, but that the recipient, or an application on the recipient demands attribute statements.

Cause value: 4

Default text string: Attribute Statements Desired Instead

An example usage in a SIP Reason header:

Reason: SAML; cause=4; Attribute Statements Desired Instead

8.3.5. Unsupported Content

"Unsupported Content" means the recipient encounters a problem with the content of the SAML assertion.

Cause value: 5

Default text string: Unsupported Content

An example usage in a SIP Reason header:

Reason: SAML; cause=5; Unsupported Content

8.3.6. Unable to Dereference

"Unable to Dereference" means the recipient cannot resolve the reference to a SAML assertion. This may mean the URI is bad, or the indicated server had some other error or rejected the request.

Cause value: 6

Default text string: Unable to Dereference

An example usage in a SIP Reason header:

Reason: SAML; cause=6; Unable to Dereference

8.3.7. Cannot Parse SAML Assertion

"Cannot Parse SAML Assertion" means the SAML assertion is not well formed.

Cause value: 7

Default text string: Cannot Parse SAML Assertion

An example usage in a SIP Reason header:

Reason: SAML; cause=7; Cannot Parse SAML Assertion

8.3.8. Conflicting SAML Assertions Supplied

"Conflicting SAML Assertions Supplied" means a recipient received more than one SAML assertion and their content is conflicting

Cause value: 8

Default text string: Conflicting SAML Assertion Supplied

An example usage in a SIP Reason header:

Reason: SAML; cause=8; Conflicting SAML Assertion Supplied

8.3.9. Insufficient SAML Statements

"Insufficient SAML Statements" means there is not enough information in the SAML assertion to sufficiently authenticate or authorize the requesting party.

Cause value: 9

Default text string: Insufficient SAML Statements

An example usage in a SIP Reason header:

Reason: SAML; cause=9; Insufficient SAML Statements

8.3.10. Dereference Timeout

"Dereference Timeout" means that the dereferencing node has not received a response within a reasonable timeframe.

Cause value: 10

Default text string: Dereference Timeout

An example usage in a SIP Reason header:

Reason: SAML; cause=10; Dereference Timeout

9. 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 16, 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 dicussions, see: Section 10.1 and Section 10.2.

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 16: Unsigned SAML Assertion Illustrating Conveyance of Subject Attribute

In the second example, Figure 17, 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>
```

```
                <ds:Transform
                    Algorithm=
          "http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
                <ds:Transform
                   Algorithm=
                      "http://www.w3.org/2001/10/xml-exc-c14n#">
                   <InclusiveNamespaces
                      PrefixList="#default saml ds xs xsi"
                      xmlns=
                       "http://www.w3.org/2001/10/xml-exc-c14n#"/>
                </ds:Transform>
             </ds:Transforms>
             <ds:DigestMethod
              Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
             <ds:DigestValue>
                Kclet6XcaOgOWXM4gty6/UNdviI=
             </ds:DigestValue>
          </ds:Reference>
       </ds:SignedInfo>
       <ds:SignatureValue>
         hq4zk+ZknjggCQgZm7ea8fI7...Hr7wHxvCCRwubfZ6RqVL+wNmeWI4=
       </ds:SignatureValue>
       <ds:KeyInfo>
          <ds:X509Data>
              <ds:X509Certificate>
     MIICyjCCAjOgAwIBAgICAnUwDQYJKoZIhvcNAQEEBQAwgakxNVBAYTAlVT
     MRIwEAYDVQQIEwlXaXNjb .....  dnP6Hr7wHxvCCRwubnZAv2FU78pLX
     8I3bsbmRAUg4UP9hH6ABVq4KQKMknxu1xQxLhpR1ylGPdioG8cCx3w/w==
              </ds:X509Certificate>
          </ds:X509Data>
       </ds:KeyInfo>
    </ds:Signature>
    <Subject>
```



```
       <NameID
         Format=
       "urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
         Alice@example.com
       </NameID>
       <SubjectConfirmation
        Method="urn:oasis:names:tc:SAML:2.0:cm:sender-vouches"/>
    </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 17: Signed SAML Assertion Illustrating Conveyance of Subject Attribute

10. Security Considerations

This section discusses security considerations when using SAML with SIP.

10.1. 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 per [I-D.ietf-sip-identity] because the imposter will not have the corresponding private key with which to generate the signed Identity header value.

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.

* does not contain an authentication statement. Thus no parties implementing this specification should be relying on SAML assertions specified herein as sufficient in and of themselves to allow access to resources.

* relying party is represented in the SAML assertion's Audience Restriction.

* Issuer is represented in the SAML assertion.

* validity period for assertion is restricted.

* etc.

10.2. 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.

10.3. Replay Attack

Threat:

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

Countermeasures:

There are various provisions within [I-D.ietf-sip-identity] that prevent a replay attack.

11. Contributors

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

12. Acknowledgments

We would like to thank RL 'Bob' Morgan, Stefan Goeman, Shida Schubert, Jason Fischl 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.

13. IANA Considerations

13.1. IANA Registration for Response Code 4XX

Reference: RFC-XXXX (i.e. this document)
Response code: 425
Default reason phrase: Bad SAML Assertion

This SIP Response code is defined in Section 8.1 of this document.

13.2. IANA Registration of the SAML Reason Protocol

The Reason Protocol value "SAML" is created by this document, with the definition and values in Section 8.1.

Cause-Code	Optional-Default-Text	Reference
Cause=1	Location Format Not Supported	[This doc]
Cause=2	Geo-location Format Desired	[This doc]
Cause=3	Civic-location Format Desired	[This doc]

Cause=4	Unsupported Schema	[This doc]
Cause=5	Cannot Parse Location Supplied	[This doc]
Cause=6	Cannot Find Location	[This doc]
Cause=7	Cannot Dereference	[This doc]
Cause=8	Conflicting Locations Supplied	[This doc]
Cause=9	Incomplete Location Supplied	[This doc]
Cause=10	Dereference Timeout	[This doc]
Cause=11	Cannot Process Dereference	[This doc]
Cause=400	Bad Request	[This doc]
Cause=403	Forbidden	[This doc]
Cause=404	Not Found	[This doc]
Cause=414	Location Error	[This doc]
Cause=500	Server Internal Error	[This doc]
Cause=501	Service Not Implemented	[This doc]
Cause=504	Server Time-Out	[This doc]

Legend:

Cause-Code - Cause value for this indication
Optional-Default-Text - optional text string of indication
Reference - document which is the reference for this cause value

14. Open Issues

A list of open issues can be found at:
http://www.tschofenig.com:8080/saml-sip/

15. References

15.1. Normative References

[I-D.ietf-sip-identity]
Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", draft-ietf-sip-identity-06 (work in progress), October 2005.

[I-D.ietf-sipping-trait-authz]

Peterson, J., "Trait-based Authorization Requirements for the Session Initiation Protocol (SIP)", draft-ietf-sipping-trait-authz-02 (work in progress), January 2006.

[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.

[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.

[RFC3326] Schulzrinne, H., Oran, D., and G. Camarillo, "The Reason Header Field for the Session Initiation Protocol (SIP)", RFC 3326, December 2002.

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

[W3C.xmldsig-core] Eastlake, D., Reagle , J., and D. Solo, "XML-Signature Syntax and Processing", W3C Recommendation xmldsig-core, October 2000, <http://www.w3.org/TR/xmldsig-core/>.

15.2. Informative References

[I-D.ietf-sip-content-indirect-mech] Burger, E., "A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages", draft-ietf-sip-content-indirect-mech-05 (work in progress), October 2004.

[I-D.ietf-sipping-certs] Jennings, C. and J. Peterson, "Certificate Management Service for The Session Initiation Protocol (SIP)", draft-ietf-sipping-certs-03 (work in progress), March 2006.

[I-D.jennings-sipping-pay] Jennings, C., "Payment for Services in Session Initiation

Protocol (SIP)", draft-jennings-sipping-pay-04 (work in progress), June 2006.

[I-D.peterson-message-identity]
Peterson, J., "Security Considerations for Impersonation and Identity in Messaging Systems", draft-peterson-message-identity-00 (work in progress), October 2004.

[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]
Hodges, J., Philpott, R., and E. Maler, "Glossary for the Security Assertion Markup Language (SAML) V2.0", OASIS Standard saml-glossary-2.0-os, March 2005.

[OASIS.saml-sec-consider-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-08]
Hughes, J. and E. Maler, "Security Assertion Markup Language (SAML) V2.0 Technical Overview", OASIS SSTC Working Draft sstc-saml-tech-overview-2.0-draft-08, September 2005.

[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.

Appendix A. Appendix: Use-case Scenarios

This Appendix explores message flows based on various use-case scenarios in [I-D.ietf-sipping-trait-authz], and from various discussions, to which SAML-based solutions are applied. Appendix A.2 shows a SIP conferencing scenario with role-based access control using SAML.

Note that we present these scenarios as illustrations of possible SAML-based use cases in SIP. This document does not provide a detailed exposition of these scenarios -- that is left for addtional documents.

A.1. PSTN-to-SIP Phone Call

Alice, using a phone connected to the PSTN, wants to make a call to Bob, who resides in a SIP network. Her call is switched through the PSTN by means of PSTN signaling (outside the scope of this document) to the PSTN/SIP gateway. At the gateway, the call is converted from SS7 signaling to SIP signaling. Since Alice's PSTN phone was previously "authenticated" via PSTN signaling mechanisms, the gateway is able to assert her phone's identity (e.g., her telephone number) via SIP Identity and SAML-based mechanisms (e.g., in order to convey profile attributes) to Bob's SIP proxy, which also dereferences the URI in the Identity-Info header in order to obtain the SAML assertion and the PSTN/SIP Gateway's domain certificate. Alice's INVITE is then forwarded from the SIP/PSTN gateway to Bob's phone, and is secured via whatever means is locally established in Bob's administrative domain.

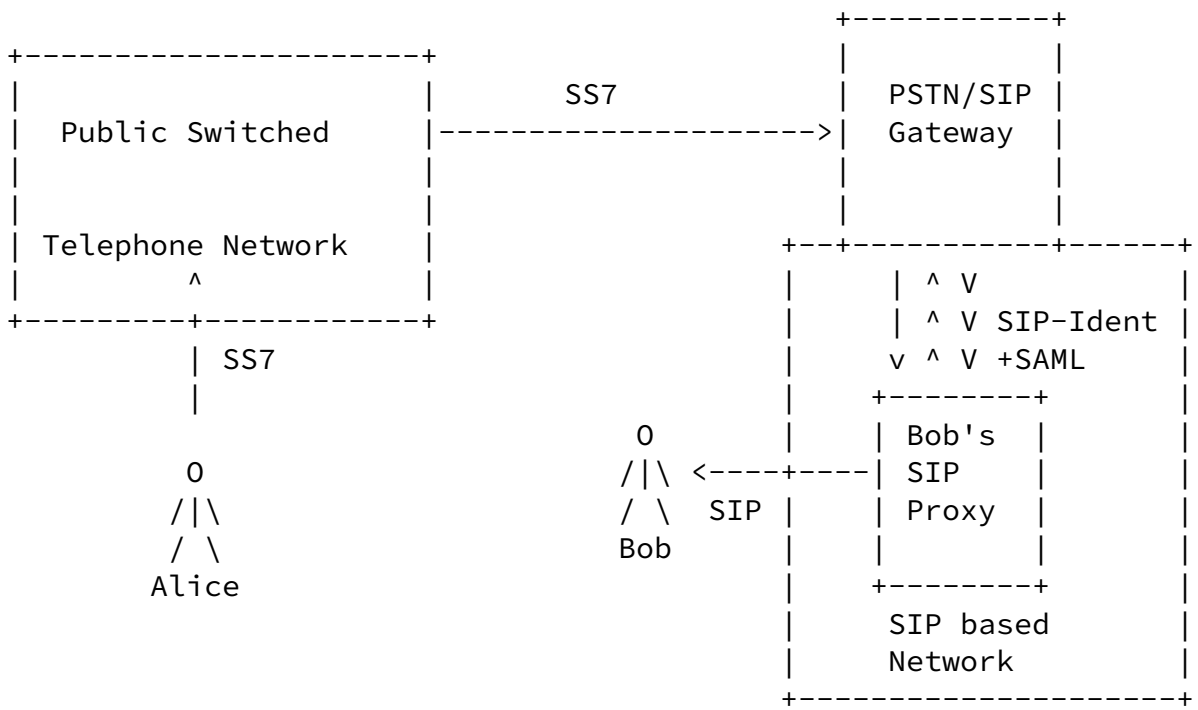


Figure 20: PSTN to SIP call

Note that the INVITE emitted by the PSTN/SIP gateway could alternatively be simply forwarded by Bob's SIP Proxy to Bob's phone, and Bob's phone could take on the SIP Identity "verifier" role, which

is being played by Bob's SIP proxy in the figure.

Whichever approach is employed is a decision local to Bob's administrative domain and can be made independently.

A.2. SIP Conferencing

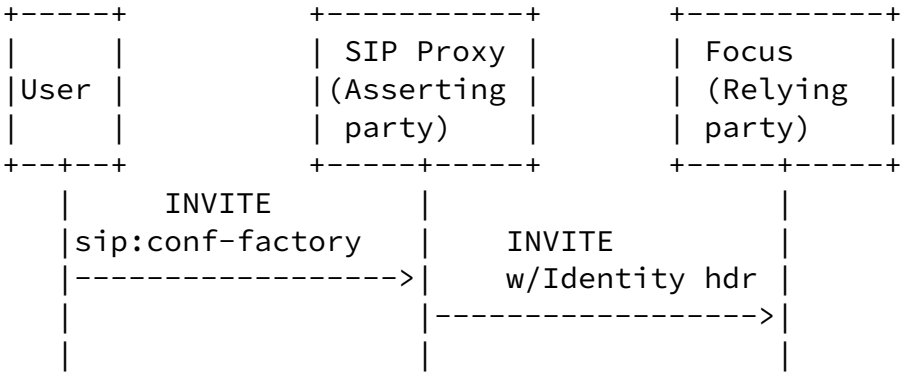
This section is meant to foster discussion about the usage of SAML in the domain of conferencing. A user agent who routes its SIP message through the Authentication Service (Asserting Party) towards a conferencing server may want or need various of her profile attributes included and may also need to be authenticated by the conference server. The following properties could be provided by this procedure:

- The user identity can be replaced to allow the user to be anonymous with regard to the Focus. This can be accomplished via [RFC3323] in combination with [I-D.ietf-sip-identity], per the latter, or,

- The user identity could be asserted to the Focus, via [I-D.ietf-sip-identity] mechanisms, and/or,

- the SAML assertion could provide additional user profile information such as group membership (belongs to the students, staff, faculty group of university X). This could, for non-identity-based authorization systems, imply certain rights.

The corresponding SIP message flow (in high level detail) could have the following shape:



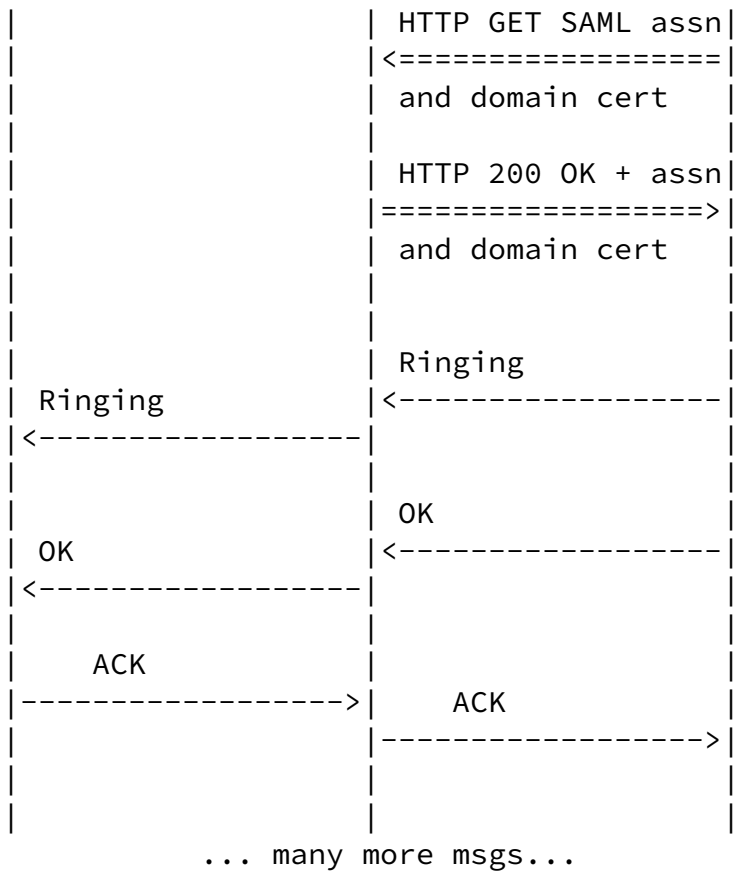


Figure 21: SIP Conferencing and SAML

However, there are obvious scaling issues with the conference server having to do the outbound requests in order to obtain SAML assertions

and certificates for conference participants.

This could be addressed by creating another SIP SAML Profile where the caller obtains the necessary information, e.g., SAML assertions, and places them into its SIP request message prior to sending it. This would obviate the need for the callee relying party to make requests in order to obtain said information. This is a topic for future work, and possibly future revisions of this specification.

A.3. Compensation using SIP and SAML

This section describes a scenario where SAML is used in SIP to

realize compensation functionality as described in [I-D.jennings-sipping-pay].

Note that this scenario is not directly addressed by the SIP SAML Profile and SAML SIP Binding presently defined in this specification. Rather, this use case calls for additional such profiles and bindings to be developed.

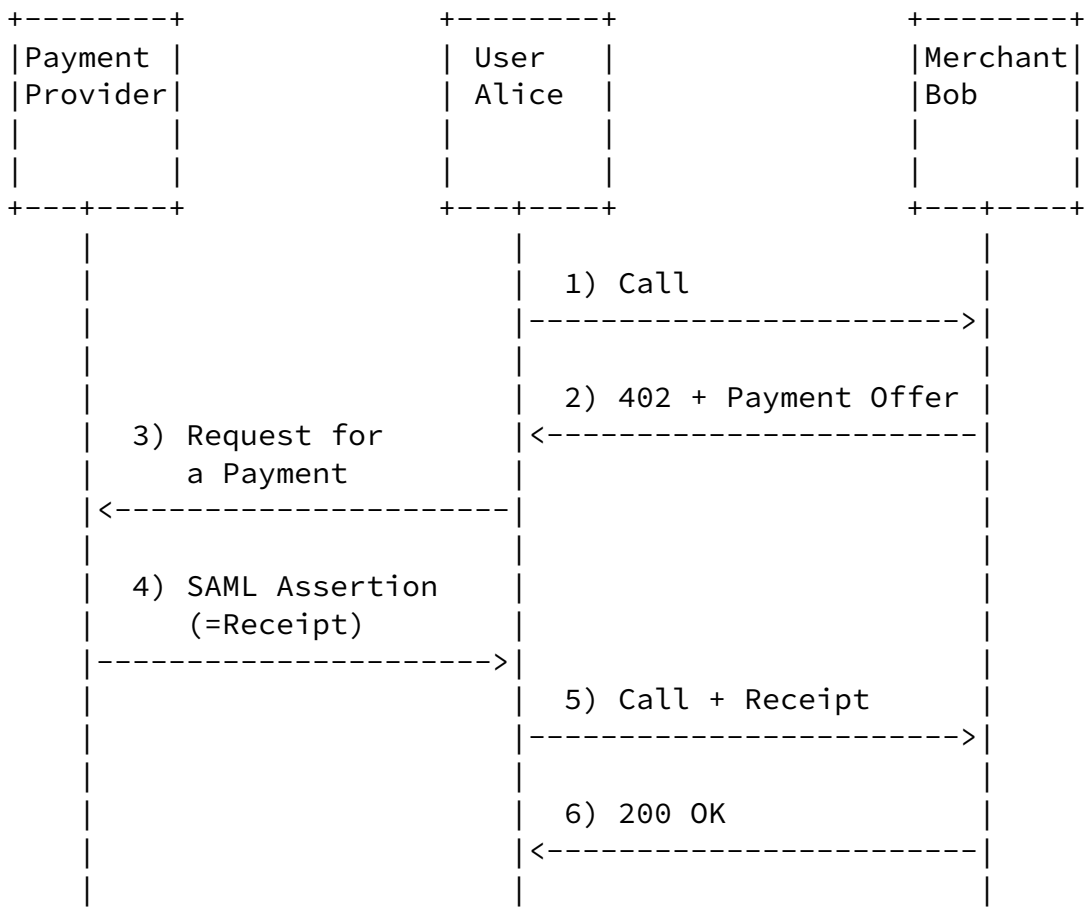


Figure 22: Message flow for SIP payment

User Alice and the Merchant Bob interact with each other using SIP and the Alice uses HTTP to exchange messages with a Payment Provider. Initially, Alice makes a call to Bob (1). Bob determines that a

payment is required and includes information about the payment in an Offer body of a 402 (Payment Required) response to Alice (2). Alice looks at this Offer and decides to make a payment. Alice therefore instructs her Payment Provider to make a transfer from Alice"s account to the Merchants"s account (3) using a request for a SAML assertion with the extensions defined in this document. The Payment Provider returns a receipt for this transfer (4). This receipt is a SAML Assertion. Alice resubmits the call to Bob but this time provides the Receipt for the transaction (5). Bob determines whether the Receipt is valid (by checking the digital signature and the content of the assertion) and continues with the call processing, if the authorization was succesful.

The Offer contains information about the participating parties (i.e., the Payment Provider, the Merchant Bob, and the user Alice), the transaction amount, the account identifier for Bob at the Payment Provider, and a replay protection indicator to make it easier for the Merchant Bob to avoid replay attacks. User Alice includes this information when making the Request for Payment to the Payment Provider; adds its own account information and authorization password; and sends this to the Payment Provider, which produces a Receipt for the transaction if it is successful. This transfer from Alice to the Payment Provider is made across an encrypted, integrity protected channel. The Receipt includes a timestamp when the Payment Provider made the transaction and protects the Receipt with a digital signature. Alice resubmits the call to the Merchant Bob with the Receipt from the Payment Provier. Merchant Bob can check for replay attacks using the timestamp and a replay protection indiciator initially provided with the Offer. Bob can then check the signature is valid using the Payment Provider"s public key.

Authors' Addresses

Hannes Tschofenig
Siemens Networks GmbH & Co KG
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

Email: Hannes.Tschofenig@siemens.com

Jeff Hodges
NeuStar, Inc.
2000 Broadway Street
Redwood City, CA 94063
US

Email: Jeff.Hodges@neustar.biz

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

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).