

# Role-Based Authorization in the Session Initiation Protocol (SIP) based on SAML

Anand Chavali

University of Colorado at Boulder  
1B40 DLC  
Boulder, CO 80309  
1-303-735-3664

anand.chavali@colorado.edu

Douglas C. Sicker

University of Colorado at Boulder  
530 ECOT  
Boulder, CO 80309  
1-303-735-4949

douglas.sicker@colorado.edu

## ABSTRACT

*This paper describes an approach to providing role-based authorization capabilities across domains for the Session Initiation Protocol (SIP). SIP defines various methods for providing authentication, confidentiality, and integrity. Authorization, however, is defined at a very rudimentary level and relies on identity alone. This paper uses role-based authorization to provide a greater and more granular level of authorization in SIP. Role-based authorization is a paradigm where authorization decisions are based on roles asserted or assumed by a user, rather than identity. We find that the use of role-based authorization in SIP allows for the expression of sophisticated authorization policies, easier management of security, and some level of anonymity. Another advantage of such a scheme as compared to identity-based schemes is excellent scalability. The approach in this paper makes use of the Security Assertion Markup Language (SAML). It expresses roles as user attributes coded into SAML assertions which are then transferred to an authorization service, which makes the decision to allow a call, reject it or flag it based on these assertions. This paper describes the mechanisms needed to incorporate SAML in SIP. It also describes an implementation of an authorization service that uses these mechanisms.*

## Categories and Subject Descriptors

C.2.6 [Computer-Communication Networks]: Internetworking – Standards.

## General Terms

Design, Security, Standardization, Languages

## Keywords

SAML, SIP, Security, role-based authorization, authorization

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

## 1. INTRODUCTION

Session Initiation Protocol (SIP) is an application layer signaling protocol created by the Internet Engineering Task Force (IETF). SIP allows entities to locate one another on a network and invite them to participate in a session.

Security is an important consideration in SIP. SIP recommends various methods for providing authentication, confidentiality, and integrity [1]. Authorization, meanwhile, is defined on an extremely rudimentary level and is based on identity alone. However, for a better authorization mechanism, information besides identity may be needed. In the words of Peterson et al., "...there are authorization requirements that are orthogonal to ascertaining identity..." [4]. In this draft [4], the need for more comprehensive authorization mechanisms is argued in some detail.

In addition to providing more information for authorization decisions, it is also desirable that the solution be scalable. This is because of the growing popularity of SIP in a variety of applications like voice-over-IP, videoconferencing, and messaging. An attribute that is always desirable is an appropriate level of granularity and the ability to express sophisticated authorization policies.

This paper proposes the use of role-based authorization for SIP. Role-based authorization is a paradigm where authorization decisions are made on the basis of a role or roles assumed by a user rather than the identity.

The solution described in this paper uses an approach that conveys user information across domains in the form of attributes on a secure, per-need basis. Such an approach would not need the information of one user to be stored in another domain, and conveying the attributes of the user would allow the user's role to be defined facilitating a more granular, role-based authorization process. To convey the security information across domains, this paper describes the use of the Security Assertion Markup Language (SAML) [6]. SAML is an OASIS specification presently used in the web space to provide single sign-on mechanisms as well as the basis for federated authorization. User attributes are coded into SAML assertions which are then transported between the SIP entities in the domains.

A key element in the usage of SAML in any framework is the definition of bindings and profiles. Bindings and profiles define ways to incorporate SAML in different communication protocols and frameworks. This paper defines two profiles for using SAML in SIP; describing the transfer of SAML assertions by value or reference. A short discussion on the requirement of bindings is presented. A security analysis of the threat model and countermeasures is also provided for each of the profiles.

This paper is organized as follows. The following section gives a brief background on the various concepts and protocols used in this paper. Section 3 provides a brief description of the architectural model for our solution. Sections 4 & 5 describe the SIP profiles for SAML and compare them. Section 6 discusses the SIP bindings. Section 7 presents a security analysis of the profiles followed by the conclusions and future work.

## 2. BACKGROUND

This section provides a short background on the various protocols and concepts used throughout this paper.

### 2.1 SIP

SIP is a protocol used for locating end points, and subsequently inviting these endpoints to a session. It operates by exchanging request messages called 'methods' and responses to these methods. A SIP network essentially consists of SIP user agents (UA) and three types of servers – proxy, registrar, and redirect. A UA that originates requests is called a UA client (UAC) and one that responds is called a UA server (UAS). A SIP call therefore is initiated by a UAC and received by a UAS. SIP strongly resembles HTTP and SMTP in certain aspects of design. While this is an oversimplification of SIP, a detailed explanation can be found in [1].

### 2.2 Role-based Authorization

Role-based authorization is a paradigm where authorization decisions are based on roles assumed or asserted by a user rather than the identity of that user. These roles could describe traits or attributes of a user such as affiliation to particular groups or could describe the role of the user in the organization or domain. For example, a given principal might be a faculty member at a university. An assertion for that principal's identity might state that they have the 'role' of a faculty member. [4]

The role-based authorization control group within National Institute of Standards and Technology has the following to say:

Role based access control (RBAC) is a technology that is attracting increasing attention, particularly for commercial applications, because of its potential for reducing the complexity and cost of security administration in large networked applications. Since the publication of the Ferraiolo-Kuhn model for RBAC in 1992, most information technology vendors have incorporated RBAC into their product line, and the technology is finding applications in areas ranging from health care to defense, in addition to the

mainstream commerce systems for which it was designed. [5]

While it is true that RBA is being applied to various problems in network security, it has not yet been applied to the area of real-time IP based systems, e.g., voice and video over IP. In real-time communication using SIP an assertion can be presented to the User Agent Server (UAS) instead of the identity of the user initiating the session. To the UAS, knowing the identity is only a means to the end of matching that identity to policies that actually depend on roles. This allows role-based authorization to offer a very compelling privacy and anonymity solution. Identity becomes one more attribute of an assertion that may or may not be shared with various destinations [4].

## 2.3 Federation and Cross-Domain Security

This section discusses federation and cross-domain security.

### 2.3.1 Federation

Network resources exist as islands, controlled and maintained by a network authority, typically a network administrator. This control of resources includes access control mechanisms in the form of authentication and authorization. A problem arises when someone from outside of a particular realm wishes to access a resource for which he/she has no authorization. Resources may be perceived as ranging from public to highly restricted, which suggests the need for granularity of access control.

One means of providing this authorization is through the development of an agreement between the user and the realm in which the resource exists. The problem with this approach is that the network authority controlling the resource must now maintain information, such as a username and password, for each foreign user. This can quickly become a burden for the network authority as the number of foreign users increase.

An alternative is to create a mutual agreement between realms, explicitly for the sharing of resources between realms. This is the federation, where access is controlled jointly by adopting certain trust agreements between realms. [3]

### 2.3.2 Cross-domain security

In order to secure such a federated model, cross-domain authorization will be needed. Cross-domain authorization entails a user in one domain being authorized by an agent in another domain. An authorization decision cannot be made by this agent without accessing the user's information. The user must trust the sharing of identifiable user information to access the remote resource. This raises several opportunities to exploit that user's privacy.

An alternative to the simple sharing of user information between domains would be to assert an attribute (e.g. authority level such as professor/researcher/student etc.) and have this attribute examined by the authority of the remote resource. The remote authority may examine the authenticity of this assertion and make a decision regarding access. Thus, delegation is practiced with each network domain in control of the information of its users. This seconds the general practice of network administrators to keep local information within the domain. This

also reduces the burden on administrators of resources that are shared across domains. They need not maintain a separate access control matrix entry for each remote user and the remote user is exposing less information about itself across a network. SAML is one example of a protocol that provides a framework for such a secure assertion of user information across domains.

A secure federated model, thus, brings together parties with common interest while offering them protection at different levels between themselves and from others. [3]

## 2.4 SAML

The OASIS group defines SAML as:

SAML is an XML-based framework for exchanging security information. This security information is expressed in the form of assertions about subjects, where a subject is an entity (either human or computer) that has an identity in some security domain. Assertions can convey information about authentication acts performed by subjects, attributes of subjects, and authorization decisions (already made) about whether subjects are allowed to access certain resources. The protocol, consisting of XML-based request and response message formats, can be bound to many different underlying communications and transport. OASIS currently defines one binding: SOAP over HTTP. [6]

Assertions can convey information about authentication acts performed by subjects, attributes of subjects, and authorization decisions about whether subjects are allowed to access certain resources. Assertions are represented as XML constructs and have a nested structure, whereby a single assertion might contain several different internal statements about authentication, authorization, and attributes. Assertions are issued by SAML authorities.

An assertion is a package of information that supplies one or more statements issued by a SAML issuer. SAML allows issuers to create three different kinds of assertions: authentication assertion, authorization decision assertion and attribute assertion.

SAML assertions/artifacts, protocol requests and protocol responses can be embedded in other structures for transport. The specification for the bindings and profiles [7] provides a framework for this embedding and transport in SOAP messages over HTTP.

## 3. SOLUTION RATIONALE

This section presents the objectives and rationale of our model and then suggests a possible solution.

### 3.1 Objectives

As mentioned in the introduction, authorization in SIP is done solely on the basis of an authenticated identity, i.e., the UAS or proxy (on its behalf) must make an 'accept' or 'reject' decision based on the UAC's identity. The principal objective here is to enhance this authorization process for SIP requests. This paper

confines itself to the authorization of INVITE requests alone, as it is the request responsible for call setup.

It is desirable to have authorization decisions that are somewhat more sophisticated than simple accept/reject decisions. Authorization decisions that are context-specific would be a significant enhancement. For instance, a policy that is sensitive to the time of the day may authorize a user only during a certain period during the day (say during business hours). For such a policy to be implemented, some form of access control mechanism is needed that can express such a policy.

One of the problems with authorization systems based on identity is that there needs to be an *a priori* establishment of privileges associated with that identity, especially when the access control policies have at least a moderate level of sophistication. Providing the authorizing entity with as much information as possible (and allowed) about the calling user will, therefore, enable a much better authorization process.

Given the proliferation of IP-based services in recent times, it is not realistic to expect that a SIP request would originate in the same domain as its recipient. This raises manageability and security issues. It is, therefore, a very important requirement for the authorization mechanism to be managed in a federated manner.

However, the federation described in section 2.2 is an identity federation. The problem with this is as follows. Consider an identity federation for the purpose described above, implemented using SAML. When a user, authenticated in one domain, wants to send an INVITE to a user in another domain, the local domain of this user will send a SAML authentication assertion to the remote domain stating the details of the authentication. The remote domain now has an assurance from the local domain about the identity of the calling user. The requirement to have more information about the user can also be met, using SAML attribute assertions. But, in order to authorize this user, the remote domain still needs to know the set of privileges associated with this user. Maintaining a set of privileges for every such user in every domain can quickly become burdensome and, consequently, infeasible.

The need is, therefore, for a federated authorization mechanism that does not rely on identity alone and also scales well.

### 3.2 Solution

The solution for this problem is to employ a role-based authorization mechanism instead of a collection of privileges for every possible user (in other words a conventional access control matrix). This will provide much more simplified management of authorization.

The two requirements for administration with a role-based authorization mechanism are the administration of roles and the setting of privileges for those roles. The latter should be done in the remote domain and the former in the originating domain. This ensures that each domain is in charge of administering its users – both for outgoing and incoming requests.

There are two more requirements to address. Firstly, the roles need to be conveyed to the remote domain. This can be done using attribute assertions. The attributes will each express a role that the user is associated with. As such the attribute authority

will use the data repository containing the association of users and roles to create attributes.

Secondly, an inherent assumption here is that the roles assignment in the originating domain is consistent with the roles that are assigned privileges in the remote domain. Ensuring this consistency is a cost, though not an expensive one for two reasons. The first reason is that the principal cost in role-based authorization is in the initial setup of privileges for roles and the administration of roles for existing users. The recurring costs are not too frequent given the general stability of access control policies and the trivial cost of role administration. The second reason is that such inter-domain collaborations and interactions are most common between similar organizations (e.g., hospitals, universities, financial institutions), which are quite likely to have similar functions, and therefore similar roles. In such cases, ensuring consistency of roles may not need much action.

The three rules of role-based authorization can be satisfied as follows. Role authentication & authorization are taken care of in the originating domain. A user who authenticates with its domain activates all the roles that it is associated with and is therefore authorized to assume. Transaction authorization is taken care of at the remote domain. Since the privileges are set according to roles, a user will not be able to access anything that is not authorized for its active roles(s).

Thus, role-based authorization when implemented with federation can provide the necessary information without causing storage concerns. Another advantage is that using role-based authorization will allow the expression of much more complex and sophisticated access control policies.

In conclusion, authorization in SIP can be improved by employing federated, role-based authorization in SIP using SAML. The next chapter defines the necessary mechanisms for this.

#### 4. MODEL

An architectural model for the mechanism described here has been presented in a previous paper [3]. This framework consists of three processes – resource registration, resource discovery, and call initiation. Of the three, call initiation is specifically where the mechanism will be used. Its relevant portions are encapsulated here.

Figure 1 provides a simplistic illustration of this model. The calling User Agent (UA) authenticates with its local domain. This authentication can use any of the authentication methods suggested in [1]. This authentication would, in all likelihood, take place during the registration process. When this user attempts to initiate a session with another user in a different domain by sending an INVITE, the proxy in the local domain intercepts this INVITE. It then initiates a process (conforming to one of the two profiles) which eventually results in the attributes (describing the roles) of the user being transferred to the remote domain in the form of SAML assertions, either directly or indirectly (depending on the profile used). The remote domain examines these assertions and makes an authorization decision on the basis of the attributes of the user. If the user is authorized to initiate the session then the INVITE is forwarded to the target UA which then responds according to [1], and in case the user is not authorized to initiate the session, the proxy responds with a

403 “Forbidden” response. On receiving the INVITE, the called UA responds with a 200 OK if it wants to set up a session with the caller and the rest of the messaging continues according to the SIP specification [1].

For the purpose of tracking local authentication, we define an abstract entity, an Authorization Service (AS), which also performs the function of generating and storing SAML assertions, packaging them into the appropriate form for transport across to the remote domain, and, on the target side, processing them to make authorization decisions (hence the name). The AS is defined as a logical entity, the physical implementation of which is left to the network architect. The recommended implementation is in the form a directory structure that interfaces with the SIP proxy, registrar, the location service (a SIP entity that contains SIP URI to IP address mappings), and probably the UA.

There is another design decision that needs to be explained. This paper describes a continuation of the work presented in one of our earlier papers [3] for the Video Middleware (VidMid) working group at Internet2. The objective there is to create a federated model to facilitate secure, web-based videoconferencing and is the chief motivation for the creation of cross domain role based authorization for SIP. The federated nature of the model necessitates some of the control in each domain be centralized to the proxy server, registrar and AS for the domain; hence, the intercept of the INVITE at the local domain.

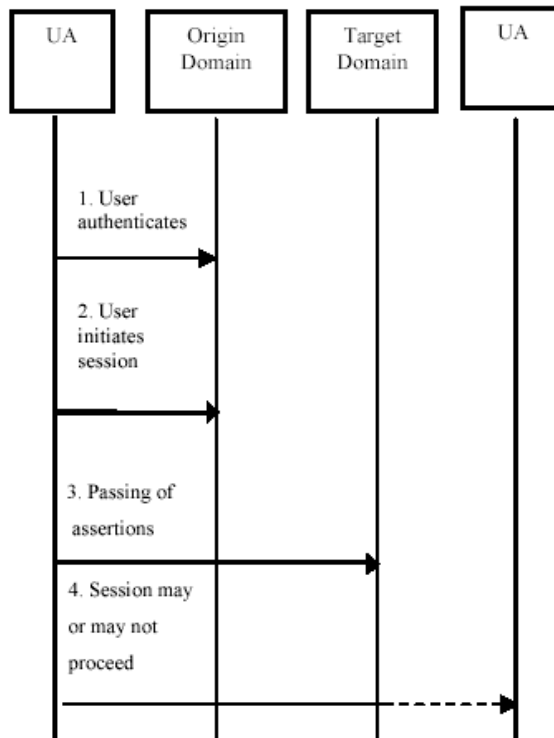


Figure 1. General Diagram

An approach that would be more in consonance with the peer-to-peer nature of SIP would involve a token being returned to the

UA by the AS. This token would then be used by the UA to initiate the assertion-transfer process, instead of the proxy. A profile based on this type of UA initiated behavior is currently being investigated. This profile would allow cross-domain, role based authorization between two UA's without any need for proxy intervention.

The model requires the identification of Policy Decision Points (PDP) and Policy Enforcement Points (PEP). As the names suggest, PDPs are where authorization decisions are made and PEPs are where they are executed. The model proposes that the AS's function as the PDPs while the proxy servers function as the PEPs in their respective domains.

Cross-domain authorization depends on an authorization service that is trusted by the User Agent Client (UAC) and the UAS. For that reason, such services are most applicable to a federated architecture where domains have agreed to trust one another's authorization services. This could be common in academic environments, or business partnerships that wish to share attributes of principals with one another [4].

## 5. SIP PROFILES FOR SAML

[7] defines profiles of SAML as, "set of rules describing how to embed and extract SAML assertions into a framework or protocol." It also describes how SAML assertions are embedded in or combined with other objects by an originating party, communicated from the originating site to a destination, and subsequently processed at the destination. Currently OASIS defines SAML bindings and profiles for SOAP over HTTP. This section defines the SIP profiles for SAML. The section discusses two profiles; the SIP Artifact Profile and the SIP Assertion Profile.

### 5.1 SIP Artifact Profile for SAML

The SIP artifact profile of SAML relies on a reference to the needed assertion traveling in the form of a SAML artifact, which the target domain must dereference from the origin domain in order to determine the user's security information. The artifact may be carried in a SIP header or as an attached MIME body.

The SIP artifact profile consists of a single interaction among three parties (a user agent, an originating domain, and a target domain), with a nested sub-interaction between two parties (the originating domain and the target domain site). The interaction sequence is shown in Figure 2, with the following paragraphs elucidating each step.

In step 1, the user agent at the origin domain sends an INVITE addressed to the user agent at the target domain, which is intercepted by the outbound proxy in the origin domain.

In step 2, the outbound proxy server of the originating domain intercepts the INVITE and responds with a '428 Artifact/Assertion Required' response adding the artifact(s) (which it obtains from its AS) to the response message as a MIME body. The status code 428 is used to indicate that "a MIME attachment or a SIP header containing SAML artifact/assertion is required" [8]. The response contains the necessary artifact as a MIME attachment or as a header.

In step 3, the artifact is extracted from the special header or the MIME type attachment of the '428 Artifact/Assertion Required' response by the UA and added to the INVITE as a MIME attachment or in a SIP header, which is re-sent. This message is intercepted by the inbound target SIP proxy server, which passes the contained artifacts to the AS in its domain.

In steps 4 and 5, the target domain (specifically its AS) dereferences the one or more SAML artifacts in its possession in order to acquire the SAML assertions that correspond to each artifact. A registered SAML protocol binding is used for a SAML request-response message exchange between the target and origin domains. The target domain functions as a SAML requester and the originating domain functions as a SAML responder.

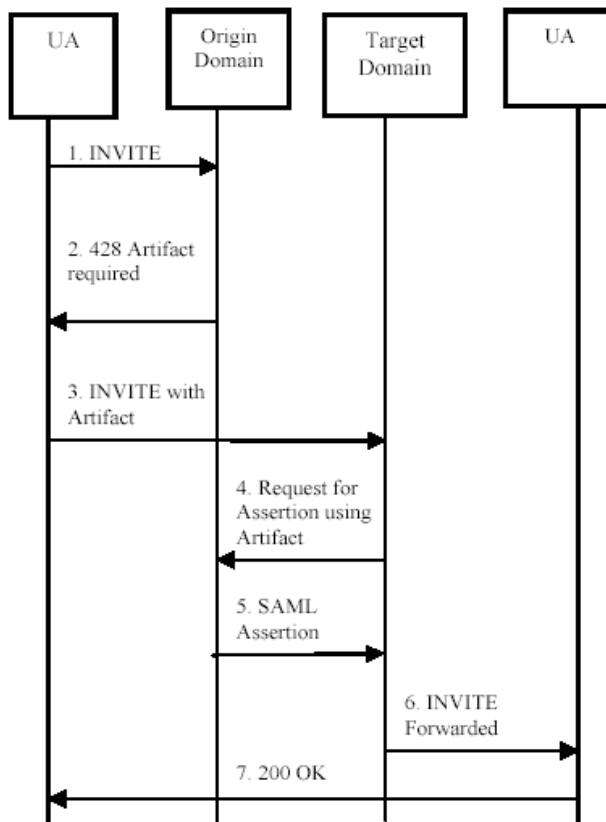


Figure 2. SIP Artifact Profile for SAML

In step 6, the origin user agent is sent a SIP response that either allows or denies access to the target user. In case the user is authorized to initiate the session the INVITE is forwarded by the inbound proxy server to the target user's UA.

### 5.2 SIP Assertion Profile for SAML

The SIP assertion profile of SAML allows authentication information to be supplied to target domain without the use of an artifact and the need for a binding

The SIP assertion profile consists of a series of two interactions, the first between a user equipped with a user agent and a source

site, and the second directly between the user and the destination site. The interaction sequence is shown in Figure 3, with the following sections elucidating each step.

In step 1, the calling party's User Agent sends an INVITE to the target user

In step 2, the outbound proxy server of the originating domain intercepts the INVITE and sends back a '428 Artifact/Assertion Required' with a MIME body attached that contains the SAML Assertion (which it obtains from the AS). Multiple assertions could be included in the MIME attachment.

In step 3, the User Agent extracts the MIME body that contains the SAML assertion/s and again sends an INVITE, with the extracted assertion as a MIME attachment.

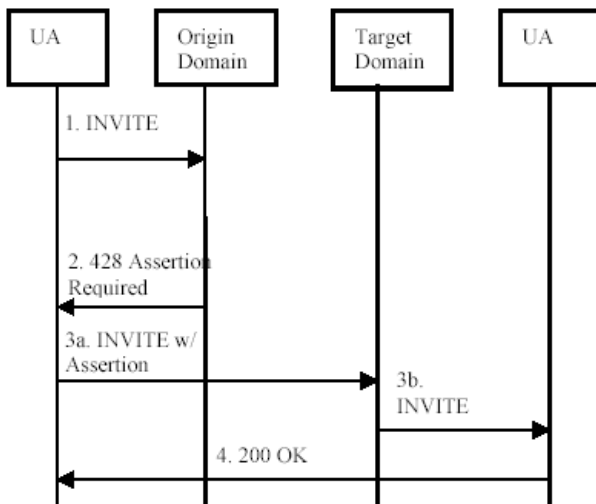


Figure 3. SIP Assertion Profile for SAML

In step 4 in case the user is authorized to initiate the session the INVITE is forwarded by the inbound proxy server to the target user's UA. The SIP response conforms to [1]. The target domain PEP can provide some form of helpful error message in the case where the originating user is not authorized to initiate a session with the target user.

### 5.3 Comparison of Profiles

The artifact profile describes a "pull" architecture, wherein the actual transfer of assertions is initiated by the target domain. The obvious advantage here is the opportunity for the target domain to request assertions describing specific attributes.

The assertion profile, on the other hand, describes a "push" architecture, wherein the assertions are "pushed" out by the origin domain. The reduction in roundtrip time is the major advantage in this case. Since the target domain does not have the opportunity to request specific attributes, the origin domain needs to know what attributes need to be sent. This will have to be specified in the trust agreement between the domains, and reduces the flexibility of the process.

## 6. SIP BINDING FOR SAML

[7] defines SAML protocol bindings as "mappings from SAML request-response message exchanges into standard messaging or communication protocols". A SIP binding for SAML would therefore describe the structure of a SIP message that would carry SAML requests and response messages. We find, however, that a binding is needed only when the artifact profile is used. In such a case, the binding would be used in steps 4 & 5. Since this communication would be between two AS's (which aren't SIP entities), the messaging protocol between them doesn't have to be SIP. Thus, we conclude that a SIP binding for SAML may not be needed. The existing binding, described in [7], can be re-used for the purpose of the AS-AS messaging.

## 7. AUTHORIZATION SERVICE

As mentioned in the section on the profiles, we have defined an abstract entity called the authorization service (AS). The AS is responsible for creating the SAML assertions and, at the other end, for processing them to make authorization decisions. This section describes a sample implementation of the AS

### 7.1 Overview

The idea of an authorization service was first introduced by Peterson, et al in a document on role-based authorization requirements for SIP [4]. It describes a framework where a UAC will send a request to an AS for an assertion, which it will receive once it has authenticated with this AS. Any suitable mechanism can be used for this authentication. Once the assertion has been provided to the UAC, it will need a mechanism of some sort to carry it. This mechanism is provided by the profiles, which are described in the previous section.

The AS implementation described here is essentially a software module that resides on the same machine as the proxy server. It runs as part of the SIP environment of the Vovida Open Communication Application Library (VOCAL) project, which is "an open-source project targeted at facilitating the adoption of VoIP in the marketplace" developed by Vovida.[11] It uses the OpenSAML libraries created by Scott Cantor of Ohio State University as part of the Shibboleth project [12]. OpenSAML is a set of open-source libraries that can be used to create, transport, and parse SAML messages.

### 7.2 Functionality

The principal functions of this AS implementation are – log relevant authentication information, create SAML authentication assertions using it, and process it to make authorization decisions based on existing policies.

**Local authentication:** The objective here is to perform local authentication and record the necessary information for use in creating the assertions. There are two UAs available with VOCAL – gua, the command-line version, and sipset, the GUI version. The latter version was used in this consideration. As soon as sipset is invoked, it attempts to register with the registrar whose details it is configured with. In the VOCAL system, a request from a UAC is always routed through a Marshal server

(VOCAL's implementation of a proxy). A Marshal server acts as a proxy for a certain set of UAs. In case the user account has been configured with a passphrase, it is challenged by the Marshal when it attempts to register. The request is forwarded to the registrar only when the challenge is successfully answered with the passphrase.

The above exchange explains the reason for co-locating the AS with the proxy. Logging authentication information in this case involves tracking the appropriate registration request and extracting the pertinent header information. Since the proxy is performing the process of local authentication in this case, locating the AS here simplifies this process considerably (especially since the other functions of the AS are fairly independent of the SIP architecture). Also, since this is the most basic implementation, the headers required for the simplest authentication assertion are chosen to be – subject, issuer, authentication method, and instant of authentication.

**Creating assertions:** The next step is to transcode the logged information into XML strings, using the Xerces library [13]. Xerces is a tool for converting to and from XML documents and is installed as a dependency along with OpenSAML. These strings can then be passed to OpenSAML library functions to create the assertions.

**Processing assertions:** Once the assertions are created, the next step involves successfully parsing them to specify decisions, which could be sent to the proxy in a variety of ways, depending on the policies in place. For instance, if the policy mandated only a clear accept or reject decision, these could be wired to the response triggers in the proxy to accordingly enforce the authorization decision. If the UAS needs more information (which could be a subset of the total information contained in the assertion), it could be provided with the INVITE in, possibly, a displayable format.

In order to implement the profiles described in the previous chapter, there are certain changes required to the SIP architecture. The development of an AS is one of them. Other changes include modification of the SIP proxy to be able to transfer control to the AS on various junctures during a call setup process – immediately after successful local authentication with the appropriate information, on receiving an outgoing INVITE without any assertion attached, and on receiving an incoming INVITE with assertions attached. There are also certain requirements on the UA – the ability to attach the assertions carried by the 428 response. These changes are being made in parallel with the AS development.

## 8. SECURITY ANALYSIS OF PROFILES

In addition to the brief security analysis provided in the previous section (and the general comments and recommendations made throughout this paper), this section provides a more in depth threat model and countermeasures analysis of the profiles.

### 8.1 Stolen or Modified Artifact/Assertion

The threat is that an eavesdropper could intercept and reuse an artifact or assertion. In general, confidentiality, bilateral authentication and message integrity must be applied across this exchange. Much of this protection comes from the use of underlying protocol security measures. Furthermore, within the message exchanges the use of accurate time stamps to provide a time-to-live feature could assist in preventing reuse.

### 8.2 Malicious Target Domain

The threat is that a malicious target domain could attempt to reuse the artifact/assertion as a source domain with some other target domain. The countermeasure relies on authentication of a source to a destination in order to obtain the assertions associated with an artifact. In other words, the site would be unable use the artifact since it does not have the appropriate origin.

### 8.3 Forged Artifact

The threat is that a malicious user could forge an artifact. This threat is addressed by the design of artifacts wherein it is infeasible to guess the value. The particular design is outside of the scope of this document but considered in [6]. In addition, measures could be taken to establish alarm thresholds on repeat requests to prevent brutal force guessing attacks.

### 8.4 User Agent State Exposure

The threat involves the storage of the artifact in some persistent memory associated with the user agent. The artifact could be stolen and reused. The countermeasure is the one-use property of the artifact as expressed in associated time stamps.

## 9. CONCLUSIONS & FUTURE WORK

This work is part of an ongoing initiative within the Video Middleware (Vid-Mid) working group of Internet2 to develop middleware functionalities in the areas of security for cross-domain multimedia applications such as videoconferencing. The first stage of this work was presented in [3]. The second stage is described in this paper and represents the most important piece in the effective and efficient securing of such applications through RBA mechanisms.

Ongoing work includes developing an implementation of the AS, which will use the profiles described in this paper to provide a complete security solution for the architecture described in [3]. Appropriate changes are also needed to the behavior of the UA and the proxy and are currently being made. Future work includes interfacing these modules to create a working test-bed and mount a series of the aforementioned attacks on it and evaluate the efficacy of each attack. A part of this research is currently being considered in the IETF as Internet Drafts within the SIPING working group.

## 10. REFERENCES

- [1] Rosenberg J., Schulzrinne H., Camarillo G., Johnston A., Peterson J., Sparks R., Handley M., Schooler E, "*SIP: Session Initiation Protocol*", Internet RFC 3261.
- [2] Ferraiolo D., Kuhn R., "*Role-Based Access Controls*", Proceedings of the 15th National Computer Security Conference, Baltimore MD, October 1992.
- [3] Sicker, D., Kulkarni, A., Chavali, A., Fajandar, M., "*A Federated Model for Secure Web-Based Videoconferencing*", IEEE Computer Society Press Proceedings of the International Conference on Information Technology: Coding and Computing, 2003.
- [4] Peterson J., Polk J., Sicker D, "*Trait-based Authorization Requirements for the Session Initiation Protocol*", SIPPING-WG Internet Draft, September 2003.
- [5] "*Role Based Access Control*", National Institute of Standards and Technology. <http://csrc.nist.gov/rbac/>
- [6] SAML 1.0 Specification Set (31 May 2002): Committee Specifications (OASIS Standard as of 5-Nov-2002)
- [7] Mishra P., et al., "*Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML)*", OASIS, May 2002.
- [8] Peterson, J., "*SIP Authenticated Identity Body (AIB) Format*" draft-ietf-sip-authid-body-01, February 2003.
- [9] Sicker D., Chavali A., Kulkarni A., Fajandar M., "*SIP Bindings and Profiles for SAML*", Work in progress.
- [10] Chavali A., "*Implementing Role-based Authorization Capabilities in the Session Initiation Protocol*", MS Thesis, Interdisciplinary Telecommunications Program (ITP), University of Colorado- Boulder.
- [11] VOCAL suite by Vovida Inc., <http://www.vovida.org>
- [12] The shibboleth project at Internet2, <http://shibboleth.internet2.edu>
- [13] Xerces-C++ library, <http://xml.apache.org/xerces-c/>