An authorization control framework to enable service composition across domains

Takashi Suzuki

University of California, Berkeley

Berkeley, CA 94720-1776

tsuzuki@eecs.berkeley.edu

Randy H. Katz

University of California, Berkeley

Berkeley, CA 94720-1776

randy@eecs.berkeley.edu

ABSTRACT

This paper contributes a comprehensive authorization control framework that enables service composition across administrative domains. The first feature of the proposed framework is a generic authorization control protocol based on SOAP/XML that can support various service components. The second is an authorization control scheme with credential transformation based on XSLT. This enables authorization decision based on local authorization rules and credentials from external domains, and liberates service providers from preparing as many authorization rules as affiliated external domains.

Keywords

Service composition, authorization control, credential transformation

1. INTRODUCTION

As service invocation over the Internet becomes a common activity, service providers are required to continuously develop new value-added services to attract users in a competitive environment. However, it is not practical for a service provider to construct every possible service. This motivates the need for service composition, where the service components on the Internet are dynamically combined at the time of execution [1].

An important issue in service composition is authorization control that can support various service components and achieve service invocation across administrative domains. The existing authorization control frameworks, however, are not suitable for this purpose. Some of them are designed for specific services, for example DIAMETER [2] for network access control. And, others are not well designed to support service provision across administrative domains. .Net Passport [3] does not consider services outside of its domain, but only those inside its realm. Kerberos [4] supports service provision access across realms, but imposes additional round trips and processing on a service user to obtain a credential from different realms.

To solve these issues, this paper proposes an authorization control framework with the following two features. The first is a generic authorization control protocol that can be universally applied to various services. Requirements for such a generic protocol are discussed in [5]. We propose a authorization control protocol, which suffices these requirements, based on SOAP/XML [6]. The second is an authorization control scheme with credential transformation. In this scheme, federated administrative domains form bilateral transformation rules for credentials based on the trust peering agreements. The authorization server converts a local authorization rule based on transformation rules, and makes an authorization decision for a service request from an affiliated domain. We designed an authorization control scheme in which the credential transformation rules are created as XSLT [7] documents to convert XML-based authorization rules.

2. Service composition models

Figure 1 illustrates an example of service composition, a content streaming system for a mobile network, where a portal provides users with appropriate content according to user profile (e.g. age, preference, location). (1) Upon receiving a user request, the portal needs to invoke back-end service components, such as (2) a user profile server and (3) a content server. (4) A content server further invokes back-end components, a content adaptation service (e.g. format conversion, content protection) and (5) a QoS management service. A content server and content adaptation service reside in different administrative domain, domain 2 and domain 3 respectively.

The certificate authority (CA) in each domain issues users and service components with certificate and secret key pairs. This pair is used for user authentication and request signing. The format of certificates is out of the scope of this paper, and we use an existing standard, such as X.509. The authorization control server issues service requestors (users or service components) with credentials. In this system, credentials

contain requestor related information such as membership status (gold, silver), rating information (official site, unofficial site), or roles. This information is described using XML-based language. The authorization control server also provides an authorization decision method to service components. Based on the authorization rule specified by the administrative domain and the service provider, the server makes an authorization decision using the credential and user profile information provided by the service component. Although there is another type of credential that contains information of accessible service (e.g. tickets), we don't use it in this system.

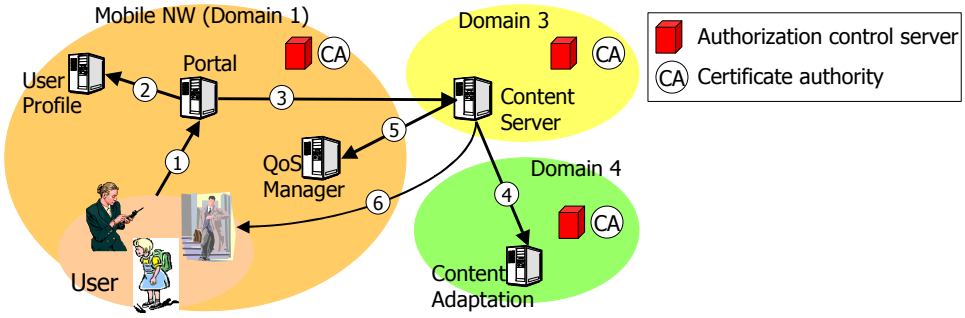

Figure 1: A customized content streaming system

3. Authorization control procedure

A service component that receives a service request invokes an authorization decision method in the authorization server using a generic authorization control protocol based on SOAP/XML. Figure 2 shows an example of SOAP message to request an authorization decision. The SOAP request and response are delivered between the service component and authorization server via HTTPS to prevent eavesdrop or falsification by malicious entities. Also, the service component and authorization server can sign SOAP messages using their secret key issued by CA as specified in XML-signature[8]. The request message of Figure 2 includes parameters for authorization decision: service, certificate, credential, profile, and condition. "Service" parameter reflects the method requested by the service requestor. "Certificate" contains a certificate sent by the service requestor. "Credential" contains a set of credentials sent by the service requestor. "Profile" contains a set of profile information obtained from user profile server. "Condition" contains a set of environmental condition information, such as server load.

When receiving the authorization decision request, the authorization server extracts the embedded parameters, and validates certificate, credential, profile, and condition by verifying the signature. If the authorization server does not have public keys for verification, it fetches certificates from the addresses specified in key information of the signature. If they are successfully validated, the authorization server input them to authorization decision method, and sends back the SOAP response containing the result.

```
Post /AuthorizationDecision HTTP/1.1
Host: www.AAAserver.com
Content-Type: text/xml; charset="UTF-8"
Content-Length: nnnn
SOAPACTION: "/AuthorizationDecision"

<SOAP-ENV:Envelope  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Header>
   <!--Header entries go here -->
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<m:AuthorizationDecision xmlns:m="AuthorizationServerInterface">
     <m:Service> … </m:Service>
     <m:Certificate> … </m:Certificate>
     <m:Credential> … </m:Credential>
     <m:Profile> … </m:Profile>
     <m:Condition> … </m:Condition>
</m:AuthorizationDecision>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 2 AuthorizationDecision Request message

4. Authorization control scheme using credential transformation

In Figure 1, a service requestor (e.g., portal in domain 1) sends a request to a service component (e.g., content server in domain 2) between administrative domains, along with its local credential. This means that the authorization server in domain 2 needs to make an authorization decision based on the credentials of domain 1. One possible way to do this is preparing as many authorization rules as affiliated

administrative domains their service invocation spans. However, it is burdensome for service providers and administrative domains to prepare many authorization rules, and could cause a security threat due to errors or omissions in the rules.

Instead, we propose a new authorization control scheme based on credential transformation. In this scheme federated administrative domains form bilateral transformation rules for credentials based on the trust peering agreements. The authorization server that receives a request with external credentials makes a decision based on the credentials, the transformation rules, and the authorization rule. There are two alternative ways to do credential transformation: applying transformation rules to credentials, and applying transformation rules to authorization rules. We chose the latter approach because it can support conditional transformation more easily. For example, if a user makes additional payments, the "gold" credential in domain 1 can be transformed to a "VIP" credential. This transformation can be achieved by just replacing the "VIP" element in the original authorization rule with the "gold" and "payment" elements.

Figure 3 shows the internal of authorization control method with credential transformation. At first, this method retrieves the authorization rule and the credential transformation rules with the key of service and the service requestor's domain name respectively. The authorization rule is converted according to the transformation rules, and inputted to the XML parser to create an authorization rule tree. At the same time, the parameters in the SOAP message, certificate, credential, and condition, are verified. If all of them are validated, they are inputted to rule tree check component to check whether the request conforms to the authorization rule.

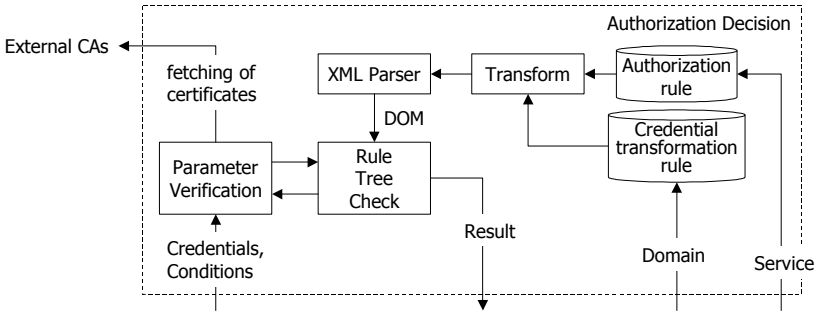

Figure 3 Authorization decision method with credential transform

5. Conclusions

To realize flexible service composition across domains, authorization control is an important issue. Challenges are to build a generic authorization control protocol that can support various services and to establish an efficient authorization control scheme that enable service invocation across administrative domains without increasing service provider's overhead to manage authorization rules.

We designed a generic authorization control protocol based on SOAP/XML so that it meet the requirements for composing various services. Also, we proposed a new authorization control scheme using credential transformation with which service providers need only to prepare authorization rules with their local credentials and transformation rules between the local credential and those of affiliated domains.

To demonstrate the effectiveness of the proposed scheme, we will implement an authorization control server with these features.

6. REFERENCES

1. Feldman, S., *The Changing Face of E-Commerce: Extending the Boundaries of the Possible.* IEEE Internet Computing, 2000. **4**(3).
2. Calhoun, P.R., et al. *Diameter Base Protocol <draft-ietf-aaa-diameter-07.txt>. IETF AAA Working Group*. 2001.
3. *.Net Passport 2.0 Technical Overview* http://www.microsoft.com/myservices/passport/technical.asp.
4. Steiner, J.G., C. Neuman, and J.I. Schiller. *Kerberos: An Authentication Service for Open Network Systems. Usenix*. 1988.
5. Farrell, S., et al., *IETF RFC2906 AAA Authorization Requirements*. 2000.
6. Box, D., et al., *Simple Object Access Protocol (SOAP) 1.1* http://www.w3.org/TR/SOAP. 2000.
7. Clark, J., *XSL Transformations (XSLT) Version 1.0* http://www.w3.org/TR/1999/REC-xslt-19991116. 1999.
8. *XML-Signature WG* http://www.w3.org/Signature.