Dynamic Trust Establishment with Privacy Protection for Web Services

Zhengping Wu, Alfred C. Weaver

Department of Computer Science, University of Virginia
151 Engineer's Way, P.O. Box 400740, Charlottesville, Virginia 22904-4740
{zw4j, acw}@cs.virginia.edu

Abstract

The lack of effective trust establishment mechanisms for web services impedes the deployment of trust models for online services. One important issue is the lack of privacy protection in trust establishment. Current web service technology encourages a client to reveal all its attributes in a standard credential to the service provider for fulfilling authorization requirements. We propose a mechanism whereby the client formulates a single trust primitive by associating a subset of required attributes in a standard credential to negotiate a trust relationship. Client privacy is preserved because only those required attributes are revealed. After negotiation, a trust group element with dynamic validation is used to represent this trust relationship.

1. Introduction

Web services have become an important media for online business services. The client of a web service and the service provider often share no prior relationship and no common security domain. Some web services require strong user authentication before allowing access to sensitive information. This paper introduces a mechanism to create a new trust relationship while simultaneously respecting a client’s privacy and keeping track of the changes within this trust relationship.

Daily life provides many examples in which a trust relationship must be established before a privilege is granted. Web services operate in an analogous way. If a student (client) attempts to open a free student checking account using an online web service, the service provider must verify the client's student status via an acceptable security token. Here, a security token is a set of claims that conveys security information within a SOAP message [1]. Only then will the service provider grant the requested service.

The exchange of security information can lead to problems. First, a credential may not be used for its intended purpose. For example, students often use driver’s license as proof of age when its intended function is to document permission to operate a vehicle. Second, a standard credential may reveal more information than is necessary. For example, a student ID may reveal academic status information (e.g., sophomore) that is not required by the bank’s web service to open an account. To solve these problems, we propose a mechanism revealing the minimal number of claims necessary to acquire the desired service to protect privacy for web service trust establishment.

2. Related concepts

SOAP is the basic infrastructure for information exchange between web services, but it does not provide any security mechanisms. Web service security is based on a process in which a web service requires that an incoming access request prove a set of claims. A web service indicates its required claims and other security related information in its policy document (e.g., a WS-Policy file). The claims are conveyed using security tokens, which are issued by a Security Token Service (STS) [1]. At the same time, attribute service maintains attribute information about principals within a trust domain.

Trust establishment is the mechanism by which one entity relies upon a second entity to execute a set of actions or to make a set of assertions. For web services, trust relationships can be built through exchanging or brokering security tokens. Thus the token-based mechanism provides a suitable solution for dynamic web service trust establishment.

3. Trust primitive element

In our proposed dynamic trust establishment framework, a trust primitive (TP) is defined as a subset of attributes signed by the attribute holder according to a set of requirements. Here, the attribute holder can be an individual user or a security domain. Every request of a TP from an outside domain will lead to a selective disclosure of attributes associated with the TP. Another merit of TPs is that they prevent initiation of selective disclosure from anyone except the attribute holder. Figure 1 shows the workflow of a round of negotiation using TPs.
(1) The client initiates a request to register the TP corresponding to the policy from the service provider. 
(2) The client’s STS registers the TP in the attribute service. 
(3) The STS adds this TP to the client’s security token. 
(4) The client embeds the security token and sends the request. 
(5) The service provider asks its own STS to check whether the request complies with the service's policy. 
(6) The provider’s STS performs attribute verification using the embedded TP. 
(7) The client’s STS uses this TP as the keyword to retrieve the corresponding attributes. 
(8) The client’s STS returns the attributes. 
(9) The provider’s STS performs verification and sends its decision (grant/deny) to the service provider. 
(10) The service provider either performs the requested operation or issues a denial. It may require several rounds of negotiation to establish a trust relationship between the client and the service provider.

4. Trust group element

After negotiation, a trust group element is added to the client’s security token or the old security token is replaced by a new one (issued by the service provider’s STS) which contains the corresponding trust group element representing the new trust relationship. A trust group element is represented by an XML tag which conforms to the WS-Security specification.

After trust establishment, requests are evaluated by checking the required trust group element. If the requirements for a trust relationship change, the service provider will invalidate the old trust group element, ask for the client's TP again, and then verify whether the new TP meets the conditions of the changed requirements. A new trust group element will replace the old one if the new TP contains the required attributes.

5. Implementation

The proposed mechanism has been implemented on the Microsoft .NET platform using WSE 2.0 SDK. All the building blocks in the implementation use web services as interfaces. We also provide a graphic user interface to assist clients with defining TPs. This trust establishment system is applied on a federated cyber trust system [2] to negotiate trust relationships. The client and web service provider establish a dynamic trust relationship via negotiation engines and security token services in both security domains. Negotiation engines control the workflow described in section 3, and the STSs use attribute services to register TPs.

6. Discussion

Seamons et al. [3] proposes some requirements for policy languages for trust negotiation, including well-defined semantics, monotonicity and credential combinations. These requirements are preserved in our proposed mechanism.

The security architecture proposed by IBM/Microsoft in [1] provides no protection in either the attribute service or the security token service that would allow the client to disclose its attributes selectively. In contrast, our mechanism better respects clients’ privacy.

7. Conclusion

In this paper we described a dynamic web service trust establishment mechanism to provide privacy control and dynamic validation capabilities. Our future work will focus on the extension of trust primitive and trust group elements to allow privacy control and protection in indirect trust establishment.

8. References

