

Federated Security: Lightweight Security Infrastructure for Object Repositories and Web Services

Marek Hatala, Timmy Eap and Ashok Shah
School of Interactive Arts and Technology
Simon Fraser University
Surrey, BC, Canada, V3T 2W1
{mhatala, teap, ashaha}@sfu.ca

Abstract

To realize the idea of web services as a scalable technology enabling access to provider's resources for a wide range of clients requires a similarly scalable security solution. A management of user accounts for all possible clients in each provider is simply unfeasible. The alternative approach to federated identity management is being developed by main software vendors. In this paper we present the design and implementation of the lightweight security infrastructure for the federated security that enables to establish a trust federation between several organizations. The infrastructure consists of supporting services and security layer on top of the web-services protocol. The solution utilizes the latest WS-Security specifications and at the infrastructure level it is compatible with Shibboleth – a federated security solution for web resources. In order to illustrate potentials of the infrastructure we describe it in the context of two case studies: object repository with complex access policies and in connection with the authenticated P2P network for learning resources.

1. Introduction

Over last few years the use of Internet as a medium where people share information to learn, play and work has increased significantly. Effective sharing and interoperability become crucial for conducting business and providing services in many sectors such as government research, education, and business in general. One of the main enablers for greater interoperability is a scalable solution for secure access to resources. The Web Services (WS)-Security group at OASIS is preparing specifications addressing this issue.

The WS-Security standard [1] defines how to send SOAP¹ messages over an insecure transport by embedding security headers that include signatures, encrypted text, and other security tokens. WS-SecurityPolicy [3] is another set of specifications that provides a standard format for specifying how web services implementations construct and check WS-Security headers. There is a substantial body of research results available in the verification of the security policies [4-6], and generating implementation code based on policies and abstract protocol descriptions [7-9] as well as tools for generating policies [6, 10].

Federated security aims to provide an identity management and secure access to resources and services among multiple organizations [11]. The Liberty Project² is a consortium over 150 organizations that “is working to address the technical, business, and policy challenges surrounding identity and web services or federated identity management”. The project develops specifications, guidelines and best practices. The Liberty identity federation framework proposes the use of federated network identity to solve the problems of federated identity [12]. The Shibboleth Project has developed an open source system that provides a federated security for web-based applications. In Shibboleth, providers make an authorization decision based on the attributes issued to the users by their home organization. The attributes and their values are agreed upon by the federation members and exchanged in the form of SAML assertions [13].

In this paper we propose and demonstrate the secure infrastructure for network of object repositories connected via web services. The infrastructure

¹ <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

² <http://www.projectliberty.org>

supports a security layer on top of the digital repository interoperability layer [14] for repository network. We defined and implemented multiple security profiles for both federated and repository-managed security. Our security infrastructure is designed with a specific goal to support an easy creation of new federations and a low-barrier adoption by new members joining the network. The connecting middleware (connector) provides a full support for security and simplifies the connection of new nodes. The connector communicates with two other infrastructure components: the certification authority and attribute authority. Similarly to the connector, both infrastructure components are easy to deploy and connect to an existing organizational infrastructure.

Our design is compatible with the Shibboleth solution by effectively extending the Shibboleth solution for web-based applications to rich clients and web services. The implementation uses the latest standards developed by the WS-Security working group.

The remainder of the paper is organized as follows. In Section 2, we present the motivation for our work and two cases that guided our work. In Section 3 we provide a background on federated security. In Section 4 we describe our security infrastructure while in Section 5 we present security profiles and analyze their security implications. Section 6 provides the discussion to the solution and a brief comparison with related work. In Section 7 we conclude and outline our future research directions.

2. Motivation

eduSource Communication Layer (ECL) is a web services based middleware infrastructure that connects learning object repositories and services in Canada, USA, UK, Australia, and Europe [14.]. The security layer for ECL would extend the repositories ability to share not only free material but also serve the material which has a certain level of access restrictions. Our design of the security layer draws on two case studies that provide the motivation for our work.

2.1 Case Study: Course Management Systems

The Course Management System (CMS) at Simon Fraser University (SFU) Surrey is used to manage and deliver courses that are structured into learning objects³. The learning objects include material developed by faculty (documents, web pages, media

documents, applets, etc.); cached copies of material from the web for which use has been cleared with the material owners; referenced materials from the digital library maintained by the SFU library with different licensing agreements with publishers; and material directly purchased from publishers for specific course offerings. To complicate the things, learning objects quite often include resources with different access rights than original resources have. For example, a unit of instruction prepared by a professor who wants to share it with the community can contain an image from an image library with a license allowing access to any member of the academic community at a Canadian university and a scanned image from a book with access purchased from a publisher for a specific group of students taking a course. Although access to these resources can differ based on the user roles such as faculty-at-sfu, faculty-at-canadian-university, general-public, undergraduate-registered-to-itec426-fall2005, etc. the current solution conservatively locks the whole repository and only students and faculty directly associated with the courses can access the material after they log in to the system.

The access to CMS for the faculty at SFU could be easily accomplished by binding the access policies to the directory information available for SFU users. However, the rest of the academic community who can have access to substantial resources through available licenses has no access to those as the attribute information is not available for the users external to SFU.

2.2 Case Study: Secure P2P Network LionShare

The LionShare project (<http://lionshare.its.psu.edu>) is developing an authenticated peer-to-peer (P2P) system that uses SAML attributes for controlling access to learning resources shared by other peers. LionShare also connects users to the object repositories by using the ECL protocol. To allow the LionShare users from different organization to access the resources in the protected repositories (such as the one described above) a solution utilizing a trust federation is needed. Our solution enables the repositories to accept attributes issued to the P2P users by their home organizations and allow them access to the resources based on the attribute values and the repository access policies. In the other direction, the repository users can use attributes issued by their organization to access resources available on the LionShare network via the ECL/LionShare bridge that converts between the protocols but maintains the security credentials.

³ IEEE Learning Object Metadata Specification defines learning object as “any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning” (<http://ltsc.ieee.org>).

3. Background on Federated Security

Different organizations have different security policies with respect to the access to their resources. Some organizations want to maintain a full control over users and their account management (e.g. Merlot). Other organizations are looking at the more scalable approach in the form of the trust federations (e.g. publishers).

3.1 Shibboleth

Shibboleth⁴ is an initiative by Internet2 working to develop security solutions that promote inter-institutional collaboration and access to digital content. It is a Single-Sign-On (SSO) and attribute-based exchange protocol consisting of two main components: Identity Provider (IdP) and Service Provider (SP). For the installation of the Shibboleth IdP, it is recommended to have a secure local authentication system and an LDAP directory storing user attributes in eduPerson organizational scheme. IdP is coupled with Attribute Authority (AA), which maintains a set of policies called Attribute Release Policies that govern the sharing of user attributes with Shibboleth SP sites [11].

In a SSO model, users sign onto their respective IdPs and can access all SPs in the federation. A *federation* is an association of organizations that use a common set of attributes, practices and policies to exchange information about their users and resources in order to enable collaborations and transactions [15]. Shibboleth achieves SSO by using Web browser technologies: redirection and cookies. When a user accesses services on an SP first time, the SP displays a list of trusted IdPs⁵. The SP redirects the user to the selected IdP with appropriate parameters necessary for IdP to validate the SP as a trusted entity and redirect the user back to the SP once the authentication is complete. The information is passed between IdP and SP in the cookies, and user's profile is delivered directly to the SP from IdP as SAML attribute assertions.

3.2 Web Services and Federated Security

Web services differ from browser transactions in that they lack the user interactions in addition to redirection and cookies. User applications access Web services on behalf of the users. The applications can determine the user profiles' attributes required to

access a service before making the request. Hence, the applications can obtain these attributes from IdP and use it to access the service on a SP. To make the operation safe a combination of security features such as certificates, signatures and encryption has to be utilized and treated in a specific way that is defined in the security profile. Our work defines the necessary infrastructure components and the profile to support the web services security in the federation of web services.

4. Security Infrastructure

The ECL Security Infrastructure (ECL-SI) is a lightweight integrated solution. It is flexible in the sense that it enables repositories to join the federation with the minimal effort. Technically, the ECL-SI allows repositories to use any IdP and to form their own circle of trust (federation). To provide this flexibility, the ECL-SI assumes that the repositories determine their own trust federation and their own access control policies while the ECL-SI provides the basic mechanisms and infrastructure components to support this trust. The ECL-SI is designed for Web services and provides three components: Certification Authority (CA), Local Attribute Authority (LAA), and Service Registry (SR).

4.1 Certification Authority

The ECL-SI solution depends on certificates to be issued to users and to repositories by trusted entities. Unlike Web application, all members including users must have a certificate signed by a trusted CA. The ECL-SI accommodates both a commercial CA and its own ECL CA. The ECL CA can be integrated into the organizational authentication system and will provide certificates to the organization members and services.

The ECL-SI supports establishing trust in a small group of organizations by organizing their CAs into a hierarchy and providing mechanism for validating the trust for all members under the same CA umbrella. It should be noted, that the strength of the security depends on the authentication system, and it is up to the federation to decide what authentication system is appropriate for their federation. The ECL CA can be integrated with a standard login module such as Kerberos (<http://web.mit.edu/kerberos/>) or simple login with username and password. However, the infrastructure also allows both repository and application developers to load and use their own login modules.

4.2 Local Attribute Authority

LAA is a component of the identity provider in the ECL-SI. Users obtain their required attributes

⁴ <http://shibboleth.internet2.edu>

⁵ Trust is maintained by the federation. For example, in the InCommon federation sponsored by Internet2 to service US higher educational institutions, all SPs and IdPs receive a list of all trusted sites and certificates to validate trust and authorize the access control to digital contents [15].

assertions from LAA to access resources at a SP. A LAA can be coupled with a CA to avoid sending authentication to two different sites. The ECL LAA is a modified Shibboleth IdP, which is adapted to work with web services. Assuming that the user applications know the required attributes to access a service, the applications formulates a Shibboleth attribute query request and calls LAA. The LAA is typically integrated with the organizational directory service where it obtains user attributes. Typically, the user certificate is also included into the SAML assertion and then the whole assertion is signed by the LAA to guarantee the attributes are presented by their owner (so called holder-of-the-key method⁶).

4.3 Service Registry

A service registry (SR) is a component that makes the infrastructure highly flexible and intelligent. The ECL SR holds information about SPs' supported security profiles, their certificates, a list of their trusted LAAs, a list of their required attributes and their values needed to access resources on the SP. Based on this information, user applications can determine if they have the required security credentials to access a resource. First, the user applications begin determining if the SPs are trusted members of the federation they associate to and then check if their LAAs are in the list. Finally, they determine if they have all the required attributes and if their attribute values match those in the list of possible values. These checks are done automatically and can be used to pre-select a set of services the application is able to communicate with. The user makes the final selection only from the list of compatible services.

In the ECL scheme, the SPs do not have to be known upfront but can be discovered by querying the registry. This is different from the other federations such as InCommon [15] where the federation maintains the list of trusted entities and the list has to be distributed to all members of the federation.

5. Security profiles

The security profile represents the description of the message exchange sequence and how it is supported by security information in the protocol messages.

The ECL-SI defines two security profiles for the web services at the level of the SOAP-based protocol. The profiles use the message level security as defined by the WS-Security standard [1]. The message level

security is achieved via combination of the security headers in the SOAP message. The headers make use of certificates, signatures and encryption of parts of the SOAP message. A combination of different security techniques within the profile allows us to achieve different levels of security. This is different from the approach used by SSL where the whole communication is over the secured wire. The messages in ECL-SI can be delivered over the unsecured wire and can be (partially) processed by the middleware services if required. ECL Security defines two profiles:

- Repository controlled security profile
- Federated security profile

In the repository controlled security profile the user has to have an account at the target repository. The users are responsible to negotiate with the repository for their access on individual basis. This profile is not the focus of this paper and more details can be found at the ECL website (<http://ecl.iat.sfu.ca>).

5.1 Federated security profile

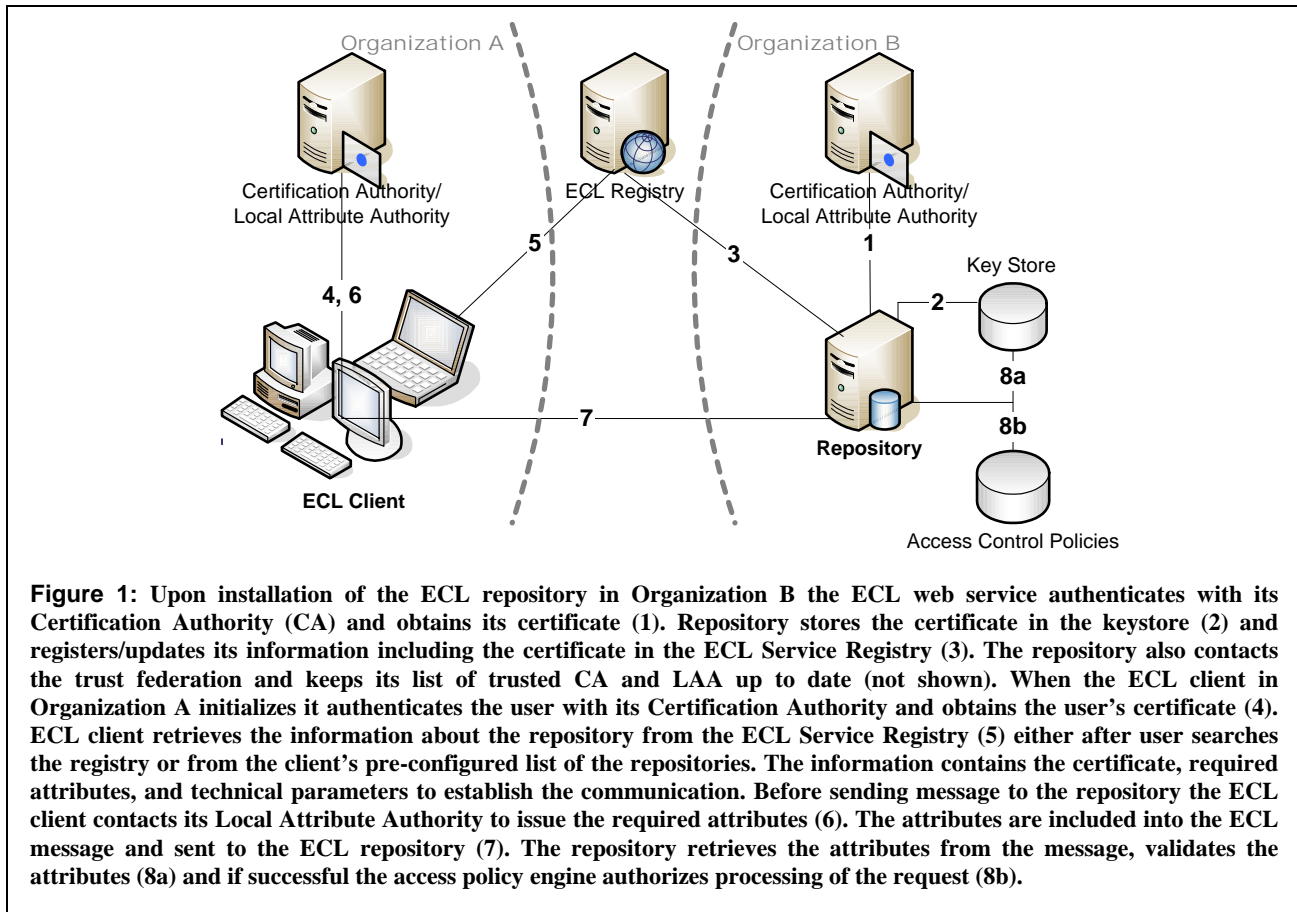
This profile supports security and privacy of communication between two parties that trust each other based on their membership to the trust federation. This means they provide access to their organizational resources based on the attributes issues by another organization.

Figure 1 shows the communication flow of messages and information passed between a client, a repository and infrastructure services. The clients discover the services via the service registry. The client user authenticates and requests attributes from his/her home institution. The repository validates the attributes and makes an authorization decision based on the user's attribute values and its access policies. As a result no user information is kept outside of their home organization.

By default the security tokens in the profile are encrypted, to avoid the possible leak of the security parameters. The ECL-SI also allows the repository to request a mandatory timestamp and signature to be included along with the security token. The signature by itself avoids the hijacking of the security parameters before the ECL message is formed. The timestamp makes it impossible to reuse the tokens after it expired.

To avoid the man-in-the-middle attack, a combination of the timestamp, signature and encryption is used. The encryption ensures that the security parameters are not readable to someone who hijacks the message. If the timestamp is both encrypted and signed any attempt to change the timestamp by a hijacker would result in the signature value not matching the modified timestamp. The combination of

⁶ The method opens the SSL connection using trusted certificate only with the holder of the matching private key. As a result the connection is established only with the agent with the proven identity.



three features makes it very difficult to reuse the security tokens.

5.2 Implementation

The ECL-SI federated profile is implemented as a middleware component called the ECL Connector. The connector uses Axis SOAP engine and WSS4J⁷ WS-Security implementation. Typically the ECL-SI profiles sit between Axis and WSS4J. The profile uses OpenSAML⁸ implementation of SAML specification.

On the sender side, the ECL connector follows the profile to prepare the security parameters and passes them to the WSS4J to prepare the secure message. On the receiver side it serves two purposes: first, it provides the required configuration for WSS4J to verify the security parameters in the message, and second, it verifies the trust and attributes received in the message according to the profile

6. Discussion and Related Work

The ECL security infrastructure has been deployed at the Simon Fraser University and can provide certificates and signed attribute assertions to SFU

users. The trust network has been created with the PennState University. As a result the LionShare and ECL users at both SFU and PennState can access resources in the CMS and P2P network using the attributes issued by their own institutions.

To the best of our knowledge similar work does not exist. However, in different aspects our work is closely related to several initiatives, including Shibboleth, WS-Security group at OASIS and Liberty Project. Our approach extends Shibboleth from the web environment into the web services environment. WS-security group concentrates on the specifications and we make use of their work. The Liberty Project [12] is much broader in scope than our work. With that respect our work can be considered lightweight with the main focus on easy setup of new trust network and low threshold enrollment of the new nodes into the trust network. Another difference is that the Liberty Project makes available only the specification while the implementation is typically not available as it is done by the commercial partners of the project. Finally, there are two other projects that attract our attention: GridShib⁹ is developing a solution for

⁷ <http://ws.apache.org/ws-fx/wss4j/>

⁸ <http://www.opensaml.org>

⁹ <http://grid.ncsa.uiuc.edu/GridShib/>

multiple organizations that wish to form a Virtual Organization or Grid. In GridShib model, SP can authenticate users (grid clients) using GridLogon and request additional users' attributes from Shibboleth AA (pull mode). In a push mode, which is similar to our approach, GridShib users authenticate and obtain attributes from Shibboleth AA and use them to make the request to SP. MAMS Project¹⁰ at Macquarie University implements inter-institutional authentication and authorization regime based on attribute exchange and XACML policies. However, the MAMS project is focusing purely on the web environment.

7. Conclusions and Future Research

In this paper we have presented security infrastructure supporting federated security for web services. The infrastructure provides the security framework for object repositories to create trust federation as well as to provide services with different levels of security. The solution defines security profiles, infrastructure services, and middleware component for a low-barrier adoption by existing repositories. Although this infrastructure can scale to large networks it is particularly sensitive to the needs of medium and small organizations, which have complex attributes and accessing policies.

The infrastructure uses WS-Security standards. It is compatible with Shibboleth infrastructure for web applications which enables developers to share infrastructure components such as attribute authority.

The infrastructure has been deployed in a pilot application to provide access to a repository of learning material with the complex intellectual property arrangements for individual resources. The access policies in the repository were based on the user attributes. The pilot successfully demonstrated the federated security for users from two different organizations.

Our current research focuses on bridging between our solution and attribute-based security that is being developed for the peer-to-peer network within the LionShare project.

8. Acknowledgements

This work was partly supported by the LionShare project funded A.W. Mellon foundation and LORNET Research Network funded by NSERC Canada. We would also like to thank Mike Halm and Derek Morr from LionShare project and to the Shibboleth working group for their input to defining the profiles. We

would also like to thank Jordan Willms for his help with implementation.

9. References

- [1] A. Nadalin, C. Kaler, P. Hallam-Baker and R. Monzillo., "OASIS Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)," Tech. Rep. OASIS Standard 200401, March 2004.
- [3] G. Della-Libera, P. Hallam-Baker, M. Hondo, T. Janczuk, C. Kaler, H. Maruyama, N. Nagaratnam, A. Nash, R. Philpott, H. Prafullchandra, J. Shewchuk, E. Waingold and R. Zolfonoon, "Web services security policy language (WS-SecurityPolicy)," December 2002.
- [4] J.D. Guttman and A.L. Herzog, "Rigorous automated network security management," *International Journal of Information Security*, vol. 4, pp. 29-48, February 2005.
- [5] A.D. Gordon and R. Pucella, "Validating a Web service security abstraction by typing," in *XMLSEC '02: Proceedings of the 2002 ACM workshop on XML security*, 2002, pp. 18-29.
- [6] K. Bhargavan, C. Fournet and A.D. Gordon, "Verifying policy-based security for web services," in *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security*, 2004, pp. 268-277.
- [7] D. Song, A. Perrig and D.E.-. Phan, *AGVI - Automatic Generation, Verification, and Implementation of Security Protocols*, 2001.
- [8] S. Lukell and A. Hutchison, "Automated Attack Analysis and Code Generation in a Multi-Dimensional Security Protocol Engineering Framework," in *Proceedings Southern African Telecommunications Networks and Applications Conference 2003*, 2003,
- [9] D. Pozza, R. Sisto and L. Durante, "Spi2Java: Automatic Cryptographic Protocol Java Code Generation from spi calculus," in *AINA '04: Proceedings of the 18th International Conference on Advanced Information Networking and Applications Volume 2*, 2004, pp. 400.
- [10] M. Tatsubori, T. Imamura and Y. Nakamura, "Best-Practice Patterns and Tool Support for Configuring Secure Web Services Messaging," in *ICWS '04: Proceedings of the IEEE International Conference on Web Services (ICWS'04)*, 2004, pp. 244.
- [11] R.L. Morgan, S. Cantor, S. Carmody, W. Hoehn and K. Klingenstein, "Federated Security: The Shibboleth Approach," *Educause Quarterly*, pp. 12-17, 2004.
- [12] J. Tourzan and Y. Koga, "Liberty ID-WSF Architecture Overview. Version 1.0," *Liberty Alliance Project*, 2004.
- [13] T. Scavo, "Shibboleth Architecture: Technical Overview," Working Draft 01, January 9, 2005, <http://shibboleth.internet2.edu/docs/draft-scavo-shib-techoverview-01.pdf>.
- [14] Reference withdrawn for blind review.
- [15] S. Carmody, M. Erdos, K. Hazelton, W. Hoehn, B. Morgan, T. Scavo and D. Wasley, "InCommon Technical Requirements and Information," vol. 2005, pp. 5, May 5, 2005.

¹⁰ <http://www.melcoe.mq.edu.au/projects/MAMS/index.htm>