User-controlled Privacy Protection with Attribute-filter Mechanism for a Federated SSO Environment using Shibboleth

Tananun Orawiwattanakul, Kazutsuna Yamaji, Motonori Nakamura, Toshiyuki Kataoka, and Noboru Sonehara
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
{tanunun, yamaji, motonori, kataoka, sonehara}@.nii.ac.jp

Abstract— Shibboleth is a well-known software package for web single sign-on (SSO) based on several federated identity standards, including the Organization for the Advancement of Structured Information Standards (OASIS)' security assertion markup language (SAML) version 1.1 and 2.0. This paper describes uApprove.jp, a user consent acquisition system (UCAS) with an attribute-filter mechanism for a Shibboleth-based SSO system. uApprove.jp requests the user's consent for the release of his/her personal information from an identity provider (IdP) to a service provider (SP) and allows him/her to determine which attributes will be sent. uApprove.jp is an extension of uApprove, a UCAS for Shibboleth. Our development is for universities participating in GakuNin (a Japanese academic federation), but it can be utilized in other Shibboleth-based federations.

Keywords-component: Japanese academic federation, user consent, Shibboleth, uApprove

I. INTRODUCTION

Currently, many academic resources, including e-journals, e-learning services, commercial content, learning management systems, portals, Grid computing, and online libraries, are in electronic format, and they can be accessed by the Internet. The term 'academic federation' in this paper refers to a federation among research and education organizations that allows students, researchers, and staff to access e-resources in multiple federated administrative domains (ADs) by initially authenticating with their home AD instead of authenticating with the accessed ADs. ADs in the federation are mutually trustworthy. User authentication is performed by his/her home AD, whereas authorization is performed by the accessed ADs. Academic federations increase the usefulness of e-resources, and they have been established worldwide, e.g., KALMAR [1] (federation among Nordic institutions), the Australian Access Federation (AAF) [2] and GakuNin (a Japanese academic federation formerly the University Public Key Infrastructure Initiative-Federation (UPKI-Fed)) [3]-[4]. GakuNin started a feasibility study in 2008. It is currently under pilot operation and should be in full operation by the end of 2010. Currently, about thirty universities have implemented the Shibboleth 2.x [5] test-bed for the security assertion markup language (SAML) 2.0 [6] federation.

A federation uses single sign-on (SSO) technologies, such as Shibboleth [5] and Kerberos [7]. When a user accesses a federated web-based service, his/her browser is redirected to his/her home AD, called the identity provider (IdP), to perform authentication, e.g., by giving a username and password. After successful authentication, the IdP sends the user's information, e.g., name and email address, to the accessed service provider (SP). The release of personal information from an IdP to an SP without the user's consent is an issue of information privacy in a federation.

Information privacy is becoming increasingly vital on the Internet. Alan Westin defines information privacy [8] as "The claim of individuals to decide whether personal information about them should be collected, processed, and used by others." Information privacy protection can be divided into two aspects: network-controlled and user-controlled. Network-controlled privacy protection includes technologies and operations performed by the service operators that provide safe data storage and data transmission, e.g., encryption and security software. User-controlled privacy protection is a mechanism that informs people on the use of their personal information, and it gives them the ability to control the release of their information.

The certification authority (CA) or commercial-certificate verification (CCV) are kinds of network-controlled privacy protection. The IdP trusts the accessed SP by using a secure mechanism, for example, the CA or CCV, and the user's data is encrypted for the transmission. Although network operators provide network-controlled privacy protection, the exchanged information is the user's personal data; As an owner of the data, he/she has a right to know the purpose of use of his/her information as well as the right to decide whether he/she will allow his/her information to be released (user-controlled privacy protection). Several schemes, e.g., the platform for privacy preferences (P3P) [9], have been proposed to provide user-controlled privacy protection by allowing the users to define their own privacy policies or attribute release policy (which information will be released to which SP). Nevertheless, this
method makes practical implementation difficult because privacy policy settings are complex and because users are not familiar with the terminology used in privacy policy settings. An alternative is the user consent acquisition system (UCAS) in which the IdP sets the privacy policy and requests the user's consent before it releases his/her information to the SP. If the user does not give consent, he/she cannot access the requested service.

In addition to providing user-control privacy protection, the UCAS implementation can also protect the IdP from future responsibility regarding violations of user privacy and data protection laws. User privacy and data protection laws vary from country to country. Currently, several federations implement a UCAS in their system, e.g., KALMAR. In Japan, the Japanese personal information protection Act (Law No. 57 of 2003) (hereinafter referred to as "Act") states that an entity shall not disclose personal information to third parties without prior consent of the individual except when it is required by law. This law has been in force since April 2005 and Japanese government entities, including national and public universities, are obligated to obey it. When collecting information, Japanese universities must inform students and staff that the objective of the collection is internal administrative use. However, students and staff should be informed about the federation’s purpose too. The UCAS requests a user's consent in real time and it also informs the user about the terms of use (ToU) and the privacy policy for an IdP, and the summary of the service.

The GakuNin development team developed a UCAS for universities participating in GakuNin. This paper presents our UCAS with an attribute-filter mechanism, called uApprove.jp. The goal of uApprove.jp is to provide of requesting a user's consent and enabling the user to control the release of his/her attributes from a Shibboleth IdP to a Shibboleth SP. Our development will be utilized in GakuNin and other Shibboleth-based federations.

Section II describes motivation behind our development and the flows in the Shibboleth/uApprove.jp architecture. Section III presents the software design of Shibboleth/uApprove.jp. Section IV presents a conclusion and discusses future work.

II. SHIBBOLETH/UAPPROVE.JP

A. Motivation for the uApprove extension

Shibboleth is a famous open source software package developed by the Internet 2 Middleware Initiative/Middleware Architecture Committee for Education (MACE). Shibboleth provides a federated SSO and attribute exchange framework. In Shibboleth, the IdP provides information about users (stored in the attribute-value pair) to the SP through the use of attribute certificates and SAML. Therefore, many organizations require a user consent acquisition (UCA) mechanism to comply with the privacy law in their countries and to inform the user about the release of his/her personal information. The current release of Shibboleth (Shibboleth 2.x) does not provide a UCA mechanism; Consequently, SWITCH, the Switzerland higher education and research network, developed a plugin-java UCAS for Shibboleth IdP 2.x called, uApprove [10]. uApprove shows the user the attributes that the IdP will release to the corresponding SP on his/her browser but he/she cannot filter some attributes that he/she does not want released (only IdP administrators can control the attribute release policy.).

People from an IdP and SP organization conduct negotiations for setting an attribute release policy. The attributes are classified into two types: mandatory and optional. The SP requests mandatory attributes for determining authorization (whether the user should be allowed to access the requested services or not.), and it requests optional attributes for providing the proper content or services that match the user's background so that the user receives a better service. An example is when a user accesses an online library; the SP requires the organization name of the user (a mandatory attribute) to determine whether the user can access the library resource or not. The SP also requests the department name to which the user belongs (an optional attribute) so that it will show the newest books regarding his/her study field on the first accessed library page. If the IdP people set the attribute release policy to release only mandatory attributes, the SP cannot provide a better service. In contrast, the IdP people could decide to release both mandatory and optional attributes to the SP; In such case, although optional attributes do not have to be released for a user, he/she has to accept their release or otherwise he/she cannot access the service. Consequently, we extended attribute-filter mechanism on uApprove. uApprove.jp requests the user's consent for release of attributes, just as uApprove does, but it allows users to filter the optional attributes that he/she is not willing to release (attribute-filter mechanism).
TABLE I: Attributes used in GakuNin

<table>
<thead>
<tr>
<th>Attribute-names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mail</td>
<td>E-mail address</td>
</tr>
<tr>
<td>sn</td>
<td>Surname</td>
</tr>
<tr>
<td>o</td>
<td>Organization name</td>
</tr>
<tr>
<td>ou</td>
<td>Organization unit</td>
</tr>
<tr>
<td>givenName</td>
<td>First name</td>
</tr>
<tr>
<td>displayName</td>
<td>Display name for a user</td>
</tr>
<tr>
<td>eduPersonAffiliation</td>
<td>Relationship to an institution, e.g., student, faculty, staff</td>
</tr>
<tr>
<td>eduPersonPrincipalName</td>
<td>ID of a user for the purposes of inter-institutional authentication</td>
</tr>
<tr>
<td>eduPersonEntitlement</td>
<td>Uniform resource identifier (URI) (either uniform resource name (URN) or uniform resource locator (URL)) indicating a set of rights to specific resources.</td>
</tr>
<tr>
<td>eduPersonScopedAffiliation</td>
<td>User's affiliation within a particular security domain in broad categories, e.g., <a href="mailto:student@nii.ac.jp">student@nii.ac.jp</a>.</td>
</tr>
<tr>
<td>eduPersonTargetedID</td>
<td>Identifier for a user shared between an IdP and SP</td>
</tr>
<tr>
<td>jasn</td>
<td>Surname (Japanese)</td>
</tr>
<tr>
<td>jaGivenName</td>
<td>First name (Japanese)</td>
</tr>
<tr>
<td>jadisplayName</td>
<td>Display name for a user (Japanese)</td>
</tr>
<tr>
<td>jao</td>
<td>Organization name (Japanese)</td>
</tr>
<tr>
<td>jaou</td>
<td>Organization unit (Japanese)</td>
</tr>
</tbody>
</table>

B. Flows in Shibboleth/uApprove.jp framework

Figure 1 describes flows in a Shibboleth/uApprove.jp framework when a user accesses a service controlled by the SP for the first time. The flows are as follows.

1. The user wants to access a Shibboleth-protected resource controlled by the SP via his/her web browser.
2. The user is redirected to a discovery server (DS), e.g., "where are you from" (WAYF). The DS is a server that contains the IdP identifiers and their location.
3. The DS shows the user a list of federated organizations. The user selects his/her home organization or IdP from the list.
4. The user is redirected back to the resource. The SP sends an authentication request to the IdP via the user's browser. Then, the user's browser is redirected to the IdP's login page as shown in Fig. 2(a). The user provides his/her login name and password for authentication.
5. If the authentication is successful, uApprove.jp displays the ToU page on the user's browser. The ToU outlines the user's obligations when using the IdP service, e.g., authentication. The ToU page is shown only when the user encounters Shibboleth for the first time or whenever ToU is updated in the time after the user last accepted it. An example of ToU page is in Fig. 2(b).
6. The user crosses a checkbox "I accept the terms of use" and clicks a confirm-button on the ToU page.
7. His/her browser is redirected to the attribute-release (AtbR) page. The AtbR page in conventional uApprove displays attributes to be released but the user cannot...
filter them. In contrast, the AtbR page in uApprove.jp displays the user mandatory attributes (without checkbox) and optional attributes (with checkboxes). The user selects the optional attributes to be released by crossing checkboxes. Note that the user can not filter the mandatory attributes. Table I shows sixteen attributes for use in GakuNin [11]. Figure 2(c) presents an example of the AtbR page, where the attributes surname and givenName are mandatory and others are optional.

8. The user can give consent by clicking the confirm-button on the AtbR web page (Fig. 2(c)). If the user clicks the cancel-button, he/she cannot access the services.

9. After receiving consent from the user, the IdP creates an SAML assertion including mandatory and user selected optional attributes, and it submits the assertion through the user browser back to the SP. The SP performs the authorization checks according to the attribute information. The user is redirected to the initially requested resource if the authorization is successful, as in Fig. 2(d). Note that if the user has authenticated and given consent to the SP, he/she can access this service later without giving consent again.

III. DESIGN OF SHIBBOLETH/UAPPROVE.JP SOFTWARE

This section describes the process of Shibboleth/uApprove.jp. Figure 3 shows the design of the Shibboleth/uApprove.jp software, and the modifications made to Shibboleth/uApprove are underlined. Note that our development was based on Shibboleth 2.1.5 and uApprove 2.1.3. Sections III.A describes the main software components, i.e., database, software package, and objects, in Shibboleth/uApprove. Our modifications of components are presented in Section III.B. Section III.C presents optional features and Section III.D describes flows in Shibboleth/uApprove.jp software.

A. Components in Shibboleth/uApprove

Here, we will briefly describe the main components in Shibboleth/uApprove software.

Shibboleth IdP: An IdP is a java web application using SAML. An IdP maintains user credentials and attributes, and it asserts authentication/attribute statements to the SP. The main components of the IdP are as follows.

Figure 3. Components in Shibboleth/uApprove.jp software
• SSO is an hypertext transfer protocol (HTTP) resource and the first accessed point. It initiates the authentication process and interacts with the authentication authority to produce the required authentication assertion.

• Authentication Authority is an SAML-defined service. It issues an authentication statement.

• Attribute Authority is an SAML-defined service that issues attribute assertions upon receiving an attribute request. It performs the attribute resolving and attribute filtering.

Shibboleth SP: An SP securely manages resources. It trusts the user authentication performed by an IdP. It makes access control decisions for a user according to the assertions issued by an IdP.

uApprove.jp: uApprove.jp is a java-plugin of Shibboleth IDP. It consists of three components: IdP Plugin, Application, and Storage.

• An IdP Plugin is a filter which determines whether the UCA mechanism should be invoked. It implements the javax.servlet.Filter interface, which is triggered when an HTTP request is forwarded to /IdP_D/profile/*, where IdP_D is the Shibboleth IdP home directory. It contains the Attribute Dumper. The Attribute Dumper retrieves the attributes, including names, values, and description, from the IdP attribute release policy so that uApprove.jp has information about what attributes will be released by the IdP.

• An Application (App) is a servlet which presents ToU and AtbR pages (java server pages (JSP) file format), and it also interacts with the user. The attributes shown on the AtbR page can be displayed in the order of the attribute listing, and attributes can be hidden. App can support different languages.

• Storage can be a Standard relational database Query Language (SQL)-based storage or a file-based storage. Our development is based on SQL storage; file storage is for a federation with a small number of users (less than 100). The uApprove.jp SQL-based storage consists of four tables: ArpUser which contains user ID, username, the last version of ToU that the user has been accepted for any SP, the first access time, and the last access time, ShibProvider which contains SP ID and SP name, AttrReleaseApproval, which contains approval ID, user ID, SP ID, timestamp, the ToU version, the requested attribute-names, and ProviderAccess, which contains the access ID, user ID, SP ID, the requested attribute-names, the ToU version, Shibhandle, and timestamp. Note that user, SP, approval, and access IDs are the unique number generated by the database system to identify each record in the ArpUser, ShibProvider, AttrReleaseApproval, and ProviderAccess tables, respectively. The ArpUser table keeps track of the latest ToU version that the user has accepted. The ShibProvider table stores the details of the SPs and it is updated by the IdP administrator. The AttrReleaseApproval table keeps records of users' consent. The ProviderAccess table keeps records of all accessed SPs including when a user's consent is not required, e.g., when uApprove.jp is in monitoring mode. If the user resets the previous approval, his/her record will be deleted from the AttrReleaseApproval table but his/her record in the ProviderAccess table still remain. uApprove.jp keeps logs in the file-based storage.

In addition to the main software components, we describe some of messages, objects, configuration data in Shibboleth/uApprove.jp as follows.

Shibboleth Authentication Request (AuthN Request):
Authn Request is a uniform resource locator (URL)-encoded message that is issued and released by the SP to the IdP for initiating a Shibboleth session either directly or via a DS. It contains several parameters in the query string, for example, the unique identifier of the SP (Generally a uniform resource identifiers (URI)), and the location of the target resource.

SAML assertion: SAML is an extensible markup language (XML)-based standard for exchanging data for authentication and authorization between an IdP and an SP. An IdP sends SAML assertions containing a packet of security information to an SP so that the SP can decide the authorization for the user. SAML assertions contain three types of statement: Authentication statements (the time and a method of authentication), Attribute statements (a name-value pair information about a user), and Authorization decision statements (what action can be performed in what resource).

Principal: It is the primary unique identifier for the user.

Login Context: it is a java object which is created by a profile handler when authentication is needed. It contains information about authentication which can be interpreted by the authentication package.

B. Modification of components for Shiboleth/uApprove.jp
We modify uApprove as follows.

Storage: In uApprove, the user cannot filter attributes and he/she gives consent for the release of all attributes. Therefore, the requested attribute-names in the AttrReleaseApproval and ProviderAccess tables is the same as the user's approved attribute-names. In contrast, these two fields in uApprove.jp can be different. Consequently, we add the field user's approved attribute-names in the AttrReleaseApproval and ProviderAccess tables. In addition, we add the fields requested mandatory and optional attribute-names into the ShibProvider table. The IdP administrator adds/updates attribute-names in the ShibProvider table.

Login Context: It is modified to contain mandatory and the user's selected optional attributes.
**Attribute Authority:** In conventional Shibboleth, the Attribute statement is set on the basis of the attribute release policy. We modify the attribute statement setting process to set the Attribute statement for the SAML assertion according to the attribute-names listed in the Login Context.

**C. Features in uApprove and uApprove.jp**

uApprove provides several optional features, i.e., the SP blacklist, ToU, Reset-approvals, and Global approval. Since uApprove.jp is extended from uApprove, it provides the same features as the uApprove does.

- **The SB blacklist** defines the accessed resources that uApprove.jp will not request for the user consent. This feature can be configured in `/IdP_D/uAppprove_D/plugin.properties`, where uApprove_D is the uApprove directory.

- **ToU** can be activated by defining in `/IdP_D/uAppprove_D/common.properties`. The ToU version and text are saved in an XML file format `/IdP_D/uAppprove_D/term-of-use.xml`. When the IdP administrator changes the ToU text, he/she has to update the ToU version in the XML file.

- **Reset-approvals** allows the user to reset his/her previous approval. The user can reset approval during the login process (a checkbox on the login form as shown in Fig. 2(a)) or the user can access the reset-approvals.jsp page.

- **Global approval** can be configured in IdP_D/uAppprove_D/viewer.properties. It allows the user to give consent for all future accessed SPs. If this feature is active, the user can give global approval by crossing a checkbox on the AtbR page (see Fig. 2(c)).

**D. Design of Shibboleth/uApprove.jp**

This subsection describes the flows in the Shibboleth/uApprove.jp software. We focus on the case in which the user gives consent. The flows in the Shibboleth/uApprove.jp software are as follows.

1. The user accesses a resource controlled by a Shibboleth SP via his/her web browser and selects his/her home organization or IdP. The SP issues and sends an authentication request (AuthN Request) to the user's IdP.

2. When the AuthN Request arrives at the IdP, the uApprove.jp IdP Plugin is triggered (Plugin Filter 1 in Fig. 3). The Login Context is null because the user has not authenticated with the IdP. The uApprove.jp IdP Plugin redirects the Authen Request to the IdP SSO, which forwards it to the IdP Authentication Authority for authentication. After successful authentication, the Authen Request is redirected back via SSO to the uApprove.jp IdP Plugin (Plugin Filter 2 in Fig. 3).

3. The uApprove.jp IdP Plug-in determines whether this request requires any authentication class that it does not provide, it is in a monitoring mode, or the SP is in the blacklist or not. If any condition is true, the uApprove.jp IdP Plugin returns the AuthN Request to the IdP and saves data in the ProviderAccess table. The IdP handles the AuthN Request according to the conventional process. In contrast, if all conditions are not satisfied, the uApprove.jp IdP Plugin checks whether the principal is unknown (authentication has error) or not. If it is, uApprove.jp shows the error, and the process is terminated.

4. Next, if the user has reset the approval, the record of the user will be deleted from the ArpUser and AttrReleaseApproval tables. Then, if the ToU feature is active, the uApprove.jp IdP Plugin checks in the ArpUser table to see if this is the first time that the user encountered Shibboleth (the record of the user does not exist in the ArpUser table) or the current ToU version (in term-of-use.xml) has been changed from the version that the user has accepted. If any condition is true, the uApprove.jp App retrieves the ToU text from term-of-use.xml and shows it on the ToU page on the user's browser.

5. After the user clicks a confirm-button on the ToU page, the record of the user is added/updated in the ArpUser table. Then, if the user has given the global consent, the uApprove.jp IdP plugin returns the AuthN Request to the IdP and saves data in the ProviderAccess table.

6. In case the user has not given the global consent, the uApprove.jp IdP Plugin checks the user's record in the AttrReleaseApproval table. If the user is accessing the SP for the first time or the current requested attributes are different from the ones the user has approved (recorded in the AttrReleaseApproval table), the uApprove.jp App shows the AtbR page on the user's browser.

7. We modified uApprove so that uApprove.jp retrieves data from the ShibProvider table to classify which attributes are mandatory and which are optional. We modified the AtbR page to contain checkboxes for optional attributes. The user can select which optional attributes they want to be released by crossing checkboxes.

8. After the user gives consent, the uApprove.jp App saves data of the user in the AttrReleaseApproval and ProviderAccess tables. The user does not need to give consent again for the future access. We modified the Login Context so that it contains mandatory and user-selected optional attribute-names. The uApprove.jp App attaches the mandatory and optional attribute-names to the Login Context. Then, the uApprove.jp IdP Plugin sends the AuthN Request back to the Shibboleth IdP.

9. The AuthN Request is processed. We modified the Attribute Authority to set the Attribute statement for the SAML assertion on the basis of the attribute-names listed in the Login Context. The IdP sends an SAML assertion to the SP.
10. The SP performs authorization according to the attributes received from the IdP. If the authorization is successful, the user can access the services. If it is not, the user cannot access the services.

Note that the flows in Step 1 to Step 6 are the same for uApprove and uApprove.jp. In uApprove, the AtbR page has no checkboxes. The user has to accept the release of all attributes for accessing the service. After the user gives consent, the AuthN Request is redirected to the IdP and the Attribute statement in uApprove is set according to the attribute release policy.

CONCLUSION

We extended uApprove to enable a user to decide which of his/her attributes an IdP can release to a SP. uApprove.jp is being tested, and it will be practically implemented in the IdPs of GakuNin universities within 2010. We are now conducting a survey of users and administrators regarding their concerns and their requirements for information privacy in GakuNin. The survey also contains questions for assessing the privacy level of each attribute from the user's aspect. The survey results will be used to build a more practical implementation.

REFERENCES