Toward Robust Pseudonymity in Shibboleth/SAML Federation against Backflow of Personal Information

Short paper

Wataru Oogami
Kyoto University
Graduate School of Informatics
Yoshida Honmachi, Sakyo-ku
Kyoto, 6068501, Japan
oogami@net.ist.i.kyoto-u.ac.jp

Takaaki Komura
Kyoto University
Institute for Information Management and Communication
Yoshida Honmachi, Sakyo-ku
Kyoto, 6068501, Japan
komura@media.kyoto-u.ac.jp

Yasuo Okabe
Kyoto University
Academic Center for Computing and Media Studies
Yoshida Honmachi, Sakyo-ku
Kyoto, 6068501, Japan
okabe@i.kyoto-u.ac.jp

ABSTRACT

The framework for Shibboleth is the web service federation which take a proper care of privacy protection. It can homologize user's real name and various information Service Provider (SP) has. That can realize because it connect information between SP and Identify Provider (IdP) by using pseudonymous ID. Such information the SP has includes user's various preferences, e.g., book tastes and a medical history. In conventional frameworks for Shibboleth, the SP has to log the usage history of users by an exchanged ID (including attributes), thus only the IdP can analyze SP's usage logs. If the IdP can look usage logs, it can touch user's privacy information. We define such flow of information as “backflow". We propose a countermeasure protocol against the backflow between the IdP and the SP by using a new ID, named hashedID . Anyone, outside of Shibboleth, cannot associate SP's usage logs with personal information directly, thus an administer of SP may meddle usage logs. In our new approach, the IdP cannot analyze the usage logs because Attribute Provider (AP) transform an ID the IdP known to unknown one, as hashedID. Additionally, we propose methods how to implement hashedID by using the realized AP model, proxyIdP and SWITCH VO. This approach enables the use of more secure Shibboleth without sharing personal information by the IdP.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Access controls; D.4.6 [**Security and Protection**]: Authentication; D.4.6 [**Security and Protection**]: Information flow controls

General Terms

Management, Security, Theory

Keywords

SAML/Shibboleth, Pseudonymity, Privacy protection

1. INTRODUCTION

As the number of web services grows, it becomes difficult for a user to manage many pairs of an ID and a password. Thus use of Single Sign On (SSO) systems attracts public attention. In an SSO system, a user can use multiple services by authentication once. The benefit of using SSO systems is to reduce the number of authentication procedures. A user doesn't have to memorize a password for an ID which is bound to a single service. Several methods of implementation of SSO are proposed, e.g., OpenID [1], OAuth [2] and SAML /Shibboleth [3]. Especially, SAML/Shibboleth can provide services without announcement a user's ID from system to service providers. Shibboleth also can sharply control for access to resources by using authorization of own attributions. This is why, we adopt SAML/Shibboleth.

Security Assertion Markup Language (SAML [4]) is a familiar authentication technology. Shibboleth [3] is a type of SAML, and implementation based on it. According to Shibboleth architecture, web services are comprised by four entities, User, IdP (Identity Provider), SP (Service Provider), DS (Discovery Service). An IdP is an entity to authenticate user's identities. In a lot of cases, the IdP equal to a certain group where the user belongs, such as university, company, and so on. An SP provides actual services, and authorizes resources or services by user attributes. A DS is a helper entity which user can find own IdP.

In the framework for Shibboleth, the IdP issue an ID as identifier of user. This ID has pseudonymity to prevent privacy leakage by aggregation of names. “Aggregation of names” is challenge to identify user by cooperating closely with some SPs and analyze logs. The SP logs a usage history by direct use of this ID. Thus, the IdP can know what the user bought, lent or edited by analyzing SP's logs. This is a problem for privacy leakage in Shibboleth on backflow. We show how we can solve the problem, a new entity set up between the IdP and the SP for transforming the ID into another one. This entity is called "Attribute Provider (AP)", and this is the standard entity of SAML. The AP realize two methods, "proxyIdP" and "SWITCH VO". We propose detailed design to implement for transformation for each method.

There are three sections in this paper below. In section 2, we describe definition a problem concern this paper. In section 3, we describe solution for backflow problem. In section 4, we describe conclusion this paper and future works in a straightforward way.

2. A PROBLEM ON BACKFLOW

2.1 Privacy information

The SP maintains a log of user's usage history by the ID in attributes to apprehend error or to prevent dishonesty. We define these logs as "usage logs" in this paper. The IdP has user's personal information as attributes in the nature of things, but it must not have usage logs because these are acute personal information. For example, book tastes,medical history, and so on. In this paper, we propose protocol to protect these privacy information against backflow.

2.2 Primary IDs of Shibboleth

There are three primary IDs on framework of Shibboleth.

Principal Name
The pseudonymity of this ID for aggregation of names is low, because the IdP issue same Principal Name for each SP.

TargetedID
This ID is typified by eduPersonTargetedID[5].The pseudonymity of this ID for aggregation of names is high. The IdP issue same TargetedID anytime for same SP. However, each SP cannot identify user even if colluding another SP.

TransientID
The pseudonymity of this ID for aggregation of names is high. This kind of ID is time-independent and SP-independent.

Especially, we enhance pseudonymity TargetedID and TransientID.

2.3 A problem on backflow

As a result, if it is happened to leak usage logs from SP at fault or incidents, it is impossible to analyze logs for outside of the Shibboleth framework, another SP or a malicious user. However, only IdP can analyze these leaked usage logs because it support equivalence between user's ID and TargetedID. We call this situation "backflow". What is risky about it, these information are able to get user's various preferences, for example, book tastes, medical history and so on.(Figure.1) For instance, according to Private Information Protection Law in Japan, usage logs do not fall into personal information because it cannot have equivalence personal information. So, usage logs have risks for sloppy management.

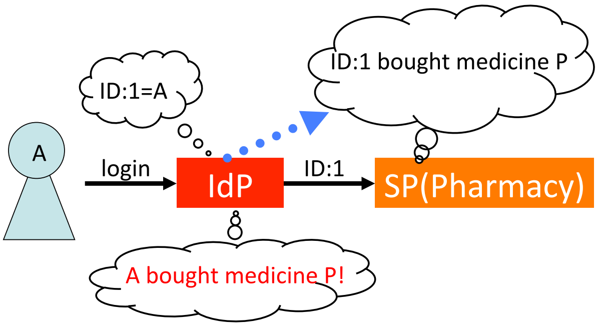

Figure 1: Example of the problem for backflow

Our goal

We set a defined our goal. Our proposal are represented by these three points.

To prevent privacy leakage from backflow
This means IdP cannot analyze usage logs by only itself

To enable to find malicious user in incident
This mean IdP can analyze usage logs by cooperation in own federation if incident is happen.

To utilize usage logs as statistical information
If the SP want to publish usage logs to someone else, pseudonymity is very important. As a matter of course, protect privacy information.

And we also implement and assess our method.

3. A SOLUTION FOR THE PROBLEM

3.1 An Approach

The most easy idea to solve the problem is that someone transform ID from issued by the IdP to another one. We named this ID "hashedID" tentatively. In order to implement this, we set up an entity between the IdP and the SP to transform the ID, such as Figure.2. We define several restrictions with hashedID.

1. The ID is given for each user, and the SP put in usage logs based on it.

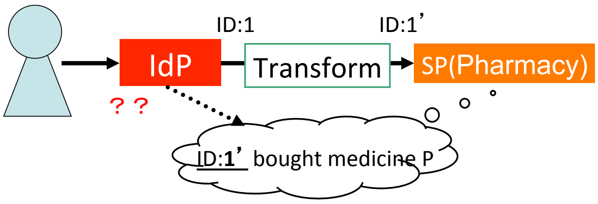

Figure 2: The idea for solving problems

2. The SP cannot associate the ID with user directly.
3. On usage logs, if backflow is happened, the IdP cannot associate the ID with the user directly.
4. If the user behave dishonestly, the IdP can identify an individual by going along with the other entities.

3.2 Attribute Provider

At first, we set up with a new entity, called an Attribute Provider (AP) to transform the ID. The AP is standard entity of SAML, so it can easy remodel. Originally, the AP is used for administering additional attributes that is more delicate than what the IdP manage. The AP falls into two principal methods.

ProxyIdP
In this way, communication between the IdP and the SP certainly through the AP. (Figure. 3) Thus, this is the most easily comprehensible method. However, it is difficult to automatize without operate by the user. In fact, the usefulness for the user is lost.

SWITCH VO
In this way, communication between the IdP and the SP is same as general Shibboleth. (Figure. 4) Instead, the SP and the AP runs in the background to exchange attributes. Thus, this method can emulate ProxyIdP and not make the user self-conscious about the AP.

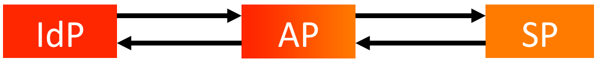

Figure 3: ProxyIdP method

These two methods have good and bad points, thus we have to consider which method fit our goal.

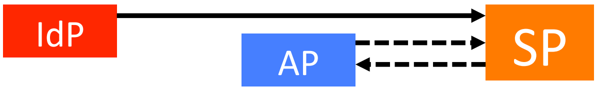

Figure 4: SWITCH VO method

3.3 HashedID

We have proposed two detailed design for implement a hashedID by using AP. It is easy way, if you realize hashedID by using proxyIdP. In this time, user have to choose destination twice, AP and IdP (Figure. 5). In ordinary framework of Shibboleth, these operation of user is only once, this is why proxyIdP is bothersome, alongside of SWITCH VO. Additionally, when incident is happened, proxyIdP can identify user intuitive way (Figure. 6). However, we consider design by using SWITCH VO method. In this case, design has to clear several restriction. In fact, The SP must not log about an ID issued from the IdP. Therefore, the ID, issued from the IdP, encrypt in advance by using method the SP cannot understand. As just described, we considered detailed design to set up the AP by using SWITCH VO method.

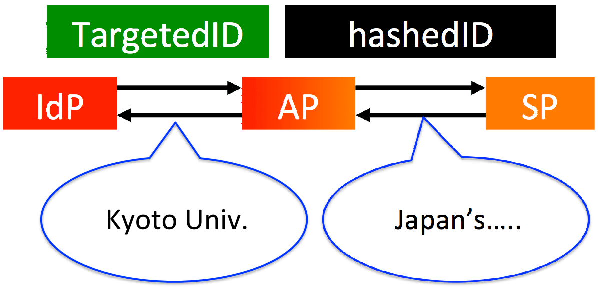

Figure 5: Example using proxyIdP

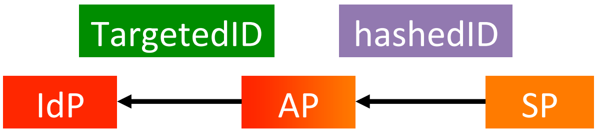

Figure 6: ProxyIdP method when an incident occurs

Access service on SWITCH VO

When user access services or resources, hashedID is issued as below (Figure. 7).

1. The IdP and the AP exchanges key (I) ahead.
2. The IdP sends transientID (Ⅱ) and encrypted TargetedID by I(Ⅲ) to the SP.
3. The SP sends Ⅲ by using Ⅱ to the AP.

4. The AP decrypts Ⅲ by using I, transforms from original targetedID or TransientID to hashedID (I V), and sends it to the SP.

5. The SP logs the usage history by using Ⅳ.

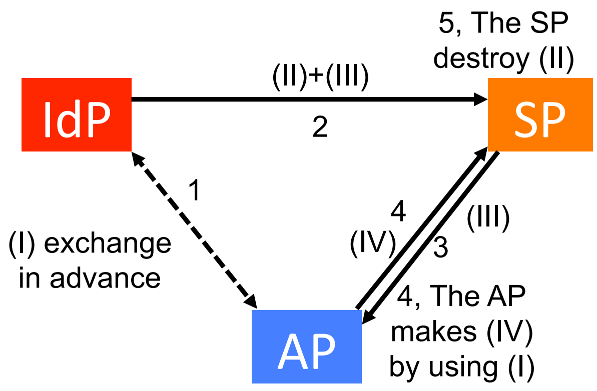

Figure 7: SWITCH VO when the user access resources

Also, it is important for this method **to abandon or to leave no log about Ⅱ**. These config prevents conjecture from external contributing factor, remaining Ⅱ on usage logs.

An incident on SWITCH VO

Administer of the SP may want to know who does the access, if incident is happened. Of course, these correspondence need policy which defined between federation in advance. If incident is happened, federation can identify user (Figure. 8).

1. The SP sends the hashedID associated with the incident to the IdP.

2. The IdP sends the hashedID to the AP.

3. The AP transforms to TargetedID or TransientID from the hashedID, and send back to IdP it

3.4 Inverse transformation of hashedID

The AP has to have methods to get original ID from hashedID, when it transform ID. There are two way.

1. The AP has an table for transformation

2. The AP transforms ID by using invertible function

In the case of 1, transform function can transform ID to anything freely. Thus, implementation is easy. However, there is a risk for leakage of the table. If leakage of the table occurs, backflow problem arise again. On the other hand, in the case of 2, the AP has to have no logs for transformation. However, the function is expected difficult, so implementation is hard. Also, the AP must conceal the function, administration is hard, too. We adopt method 1, because implementation is easy. Of course, we also check up method 2.

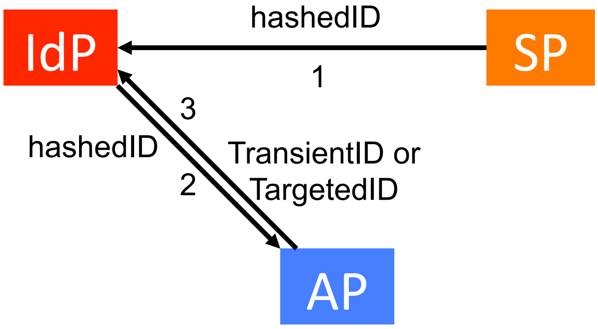

Figure 8: SWITCH VO when an incident occurs

4. CONCLUDING REMARKS

In this paper, we have revealed the problems of privacy disclosure overlooked so far in backflow in conventional Shibboleth. Then, we propose a more secure countermeasure protocol against backflow by using AP and hashedID. We are now implementing the protocol and we will evaluate it.

5. REFERENCES

[1] OpenID Foundation. OpenID. `http://openid.net/`, Jun,2007.

[2] OAuth Core Workgroup. OAuth 1.0a. *OAuth Core 1.0 Revision A*, 24 June,2009.

[3] Shibboleth-A project of the Internet2 Middleware Initiative. `http://shibboleth.internet2.edu/`.

[4] Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0. *OASIS Standard*, 15 March 2005.

[5] Keith Hazelton(The editor of the MACE-Dir working group). eduPerson Object Class Specification (200806). *Internet2 Middleware Architecture Committee for Education, Directory Working Group (MACE-Dir)*, 30 June 2008.