PUSH IDENTITY; Usable Single Sign-on

Kyohei Furukawa
Kanagawa Institute of Technology
Shimo-ogino 1030 Atsugi-shi Kanagawa, Japan

Manabu Okamoto
Kanagawa Institute of Technology
Shimo-ogino 1030 Atsugi-shi Kanagawa, Japan
manabu@nw.kanagawa-it.ac.jp

Abstract—**Single Sign-on is not yet effective. All portal service servers want to be Identity Provider and users need to have each Identity of the portal servers and it is NOT single sign-on. In this paper we propose new method of single sign-on. Users can choose one of much IDs and use it freely to do single sign-on.**

Keywords- Single Sign on, Authentication, Identity

I. INTRODUCTION

Single sign-on (SSO) is a mechanism whereby a single action of user authentication enables a user to access web services without needing to enter multiple passwords. The standard technique of SSO is either SAML [1] or OpenID [2].

Both of these techniques employ Identity Provider (IDP). Users are authenticated by IDP once, and when the user accesses the Service Provider (SP), IDP issues a SSO assertion to the SP and the SP then confirms the user's identity.

In this scheme, all users need to visit IDP in order to use the SP; hence, IDP always has a dominant business advantage. Because many portal sites want to act as an IDP, users eventually need to use plural IDPs concurrently. This is not an effective approach because when a user wants to use multiple SPs, he may need to remember multiple passwords for all IDPs. This is not a "single" sign-on.

In this paper, we propose "Push Identity Single Sign-on". When you log-in one SP, you can single sign-on any other SPs using the identity which pushed on common space in shadow IDP by the SP you logged-in first.

II. RELATED WORK

Single sign-on is standard technique of user identification. For example, OpenID is an open standard. When a user uses a service provider (called Relying Party (RP) in the OpenID glossary), RP redirects the user to IDP (OpenID provider (OP)) and OP authenticates the user and directs him back to OP with an authentication response. OP then confirms the response and identifies the user.

SAML performs almost same action as OpenID. In particular, the redirect action that IDP performs for the user is same. All users have to visit IDP (or OP) and log in before using SP. IDP must be a portal site. IDP can act as an identity service, and thus it can obtain a business advantage. For example, IDP can display advertisements at the top of the page and all users using SSO have no choice but to view these ads. This is a problem in the single sign-on scheme.

All portal sites want to be IDP and when users want to use many SPs , as a result users have to use some IDP concurrently.

When user Alice want to SP-a she have to login IDP-a. Next when Alice want to move to SP-b she have to login IDP-b. Because both IDP-a and IDP-b want to get advantage of portal site, such situation occurs.

III. WHY PUSH IDENTITY

Here, we propose a new single sign-on scheme using 'Push Identity,' a scheme that makes it possible to create a single sign-on without IDP. It gives all SPs an equal opportunity to act as an IDP, and none of them receive any special business advantage. While this scheme actually does have IDP, its users do not need to be aware of this. In this scheme, IDP is only just common space to stock the identity where SP pushes the user's identity.

Figure 1 shows existing scheme like OpenID or SAML. First users login IDP and make identity on IDP. Next when users access to SP and use sing sign-on, they PULL identity from IDP. SP confirm the ticket which shows identity on IDP. We call this system 'Pull Identity' type.

Figure 2 shows our new scheme. If you log in the SP somewhere once, you can use any other SPs freely by SSO.

In figure 2 when the use login SP first , SP PUSH the identity for common space. This common space is IDP substantially but users are NOT aware of it. So we also call this common space 'Shadow IDP'. Identity which one SP pushed is used for any other SPs. For example when user access to SP2 and use single sign-on, SP2 pull the identity which SP pushed.

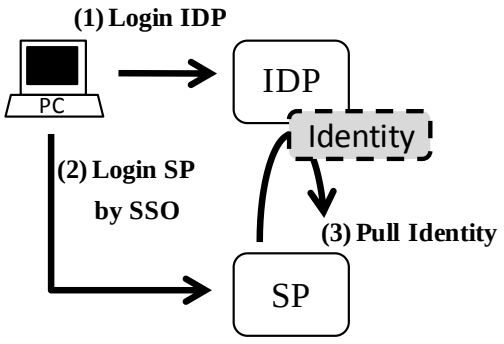

Figure 1. Pull Identity.

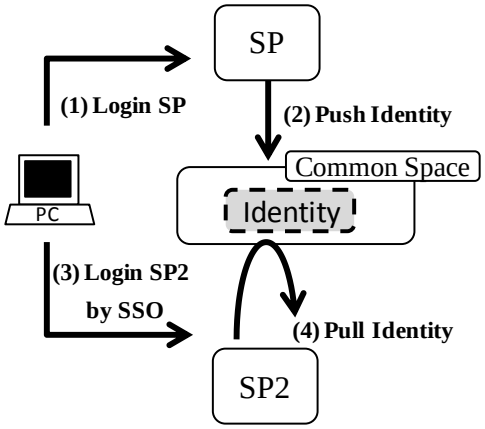

Figure 2. Push and Pull Identity.

Here, we list all the steps in this scheme, which are illustrated schematically in Figure 3. Actually we developed this system with OpenID. So we assume that this sequence almost consists of OpenID steps.

1) User Alice decides to use SP and clicks a SSO button such as "Single Sign-on here!"
2) SP redirects Alice to IDP (Shadow IDP).
3) IDP checks whether Alice has logged in to IDP. For example, IDP may check her cookies.
4) If Alice has not yet logged in to IDP, IDP redirects her back to SP.
5) SP says: "Please log in here. Please enter your ID/Password to this site."
6) Alice enters the ID/Password for this SP.
7) If SP confirms the ID/Password, it permits Alice to use the site.
8) SP then pushes the information that "Alice has already logged in to our site" to IDP.
9) IDP recognizes that Alice has already logged in. IDP may create a cookie.
10) IDP redirects Alice back to SP.
11) SP lets Alice know that SSO is complete.
12) Alice is able to use SP.
13) Alice decides to use another SP (SP2) with SSO.
14) SP2 redirects Alice to IDP.
15) IDP checks whether Alice has already logged in, for example by checking her cookies.
16) When Alice has already logged in to IDP, IDP issues an SSO assertion and redirects Alice back to SP2 with that assertion.
17) SP2 confirms the assertion and grants access to Alice.

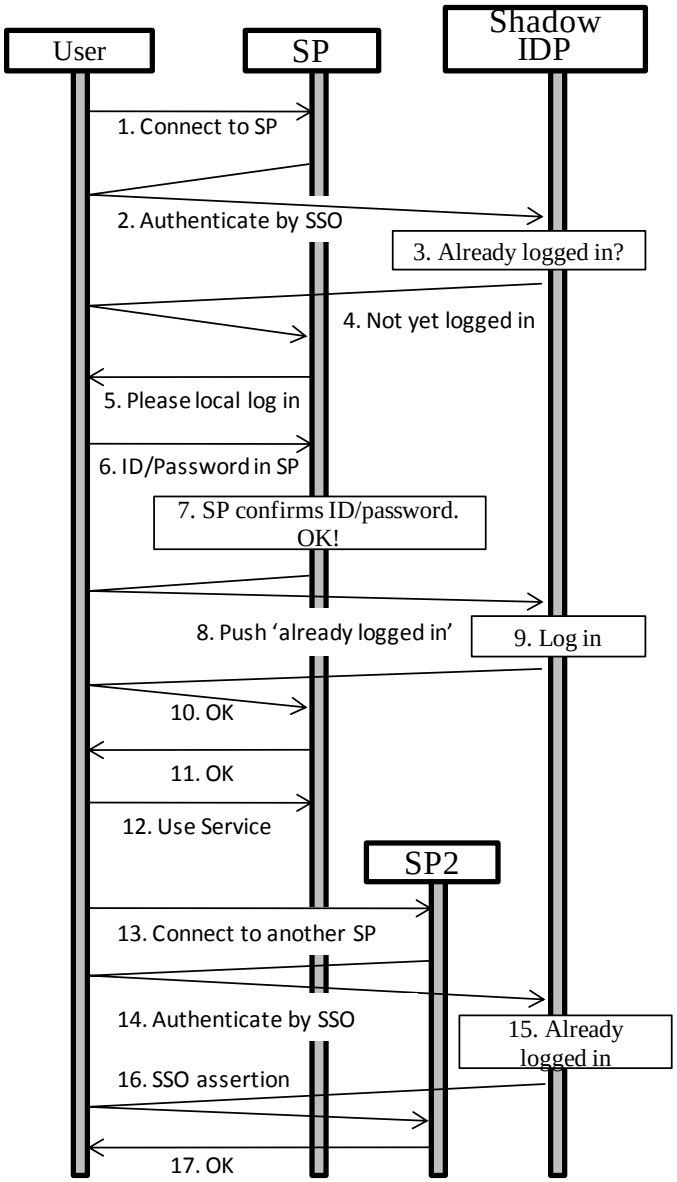

Figure 3. Sequence.

IV. CONCLUSION

In this paper, we proposed a new single sign-on scheme without needing to be aware of IDP. Users can begin from any SP and can use SSO freely. In this scheme, the initial SP to which a user logs in pushes the identity to shadow IDP, which then provides that identity to other SPs.

In this proposal, all SPs can equally act as an identity service and users can do SSO. In the future, we need to consider how to federate accounts of all SPs accounts easily.

REFERENCES

[1] OASIS Security Assertion Markup Language (SAML), http://www.oasis-open.org/committees/security.
[2] OpenID, http://openid.net/.