Poster: OpenID$_{email}$ Enabled Browser, Towards Fixing the Broken Web Single Sign-On Triangle

San-Tsai Sun
University of British Columbia
Vancouver, Canada
santsais@ece.ubc.ca

Kirstie Hawkey
University of British Columbia
Vancouver, Canada
hawkey@ece.ubc.ca

Konstantin Beznosov
University of British Columbia
Vancouver, Canada
beznosov@ece.ubc.ca

1. INTRODUCTION

Today's Web is site-centric; a user has to maintain a separate copy of their identity and corresponding password for each content-hosting and service provider (CSP). A large-scale study of password habits found that a typical web user has about twenty-five accounts that require passwords and types eight passwords per day [2]. Web users face the burden of managing this increasing number of accounts and passwords, which leads to "password fatigue". Aside from the burden on human memory, password fatigue may cause users to devise password management strategies that degrade the security of their protected information.

Web single sign-on (SSO) systems are meant to address the root causes of the password fatigue problem [1]. A Web SSO system separates the role of identity provider (IdP) from that of relying party (RP) to enable users to leverage one identity across multiple RPs. An IdP issues identities or credentials to users, while an RP depends on the IdP(s) to assert the users' credentials before allowing them access to its services. In addition to reducing users' memory burden, a globally adopted Web SSO solution can enable content sharing across and beyond the boundaries of CSPs [15].

OpenID [13] and InfoCard [16] are mainstream Web SSO solutions targeted for Internet-scale adoptions; however, they are facing RP adoption problem. Evidence shows that although major CSPs acted quickly to become OpenID IdPs (over one billion OpenID-enabled accounts), only a limited number of websites have adopted the role of RP [11, 4, 9, 10]. This is similar to having more than a billion keys, but few locks in which to to use them. For InfoCard, the list of RPs and IdPs is almost empty [16].

Fundamentally, Web SSO systems shift the functions of identity collection and authentication from RPs to IdPs. However, the incentive for RPs to rely on the identity assertion services provided by IdPs is insufficient. RPs are not willing to relinquish control over their user base unless they can obtain user data and verify the IdP's authentication and data collection policies [5, 1]. In addition, RPs have to choose which IdPs to trust as they are liable for the loss when IdPs get compromised [8].

Current Web SSO solutions do not provide RPs with sufficient business incentives to support Web SSO for users. CSPs are reluctant to modify their login UI and process because new login procedures might confuse and upset users [3, 14]. In addition, RPs might not want to expose their users to potential business competitors because once the attention of users has been redirected to an IdP during the login process, they might not return [1]. As early adoption would not provide RPs with competitive advantages, CSPs would rather wait until Web SSO technology is mature and the cost of user training has already been absorbed by other websites.

To encourage adoption by RPs, Web SSO systems have to rely on the demand from users as the driving force. However, the interaction flows provided by today's Web SSO solutions are *shared-identity* sign-on (SISO) ones rather than true *single* sign-on. With SISO solutions, users can use one identity to sign into multiple RPs. Nevertheless, when accessing N RPs using one IdP, the user must visit $N + 1$ different login forms (one for each RP website and one on the IdP), choose an IdP to login N times via N possible ways, read the consent page (e.g., consent to release identity attributes, setup a custom identifier) on the IdP N times, and log out $N + 1$ times through $N + 1$ different interfaces. These complex and inconsistent user experience imposes a cognitive burden on web users [3, 14, 1].

SISO redirects the user's attention during the login process. However, in addition to usability issues [3, 1], redirection exposes users to phishing attacks [1, 7] and makes IdP tracking possible [6]. InfoCard redirects users only to their identity selector. However, for N RPs using one IdP, the user has to provide her credential to the IdP N times unless a self-issued card (without password protection) is selected. Moreover, using multiple identities in one browsing session complicates the process for users even further. When users sign on with multiple IdPs in one browser session, they have to remember which identity was used for accessing which RP. Mixing identities in one browser session can make it difficult for users to determine why an access failed and who to contact when a problem is encountered.

SISO requires usable interfaces and flows from both RPs and IdPs. Simply adding an OpenID textbox or InfoCard icon to the traditional login page is not an option [3, 14]. To improve the user experience, some OpenID RP adopters provide a list of IdP logos on their login form for users to choose from. The users can simply click on an IdP icon to initiate a sign-on process. However, this approach leads to the "NASCAR" problem [7] when the list of IdPs grows too long to fit on the login screen. In addition, using an IdP-list restricts users' freedom of choice, which impairs healthy competition in the ecosystem.

SISO is especially problematic for Web 2.0 applications that require access personal data located on multiple CSPs. For OAuth-based applications or server-side mashups that process a user's personal content from different providers, seeing a login form on each CSP is annoy and imposes a cognitive impact on the user. For client-side mashups that use Ajax-style web services to acquire user data from several websites, login forms will block such communications. In addition, SISO-based solutions are difficult to use on mobile devices that have limited input capabilities.

2. APPROACH

Our research goal is to develop a Web SSO solution that requires *minimal* user interaction and provides RPs with clear value propositions to motivate their adoption. In our vision of a true Web SSO system, a user should log into her IdP once and gain accesses to all websites that she has an account with, *without being prompted to login again on each website*. In other words, when accessing N RPs using one IdP, the user should provide her credential exactly once to the IdP, consent at most N times (one for each RP if the consent was not recorded for future use), and should perform a logout process only once from the IdP.

Designing an usable Web SSO solution that fulfills our vision and motivates RPs' adoption is challenging. To be usable, the solution must leverage the skills and experiences that an average Web user already has. It must not require any special software being installed on end-user computers or require users to manage public/secret key or X.509 certificates for performing cryptographic operations. There are currently over one billion OpenID-enabled "keys" provided by major CSPs [10], and they are too valuable to be ignored. Thus, the solution must be backward-compatible with existing IdPs and RPs and support gradual adoption. To facilitate RPs' adoption, the solution must not require RPs to modify their login UI; how to design a usable login UI and flow for web users is still an open problem that ongoing single sign-on research is attempting to address [14, 12]. Furthermore, the solution should assist RPs in improving the *conversion rate* (the percentage of website visitors who become registered users) on their websites to motivate their adoption.

To fix the broken Web SSO triangle, we argue that it is time to build identity support directly into web browsers. We propose a new approach for Web SSO that leverages OpenID and email, and builds identity support into browser. Our approach (1) builds OpenID support directly into web browsers, (2) hides OpenID identifiers from users through the use of their existing email accounts, (3) extends the OpenID protocol to perform authentication directly with user-agents such as browsers (an OpenID$_{ua}$ extension), and (4) introduces an `OpenIDAuth` HTTP access authentication scheme to convey authenticated identities automatically to websites that support OpenID for authentication. To evaluate the feasibility of our approach, we implemented all proposed protocols in Java and set up an OpenID$_{email}$-enabled IdP and five RPs. In addition, we designed an OpenID$_{email}$ Firefox extension to demonstrate our vision of a true Web SSO solution.

With our approach, web users authenticate with their existing email accounts via an OpenID$_{email}$-enabled browser. With the user's consent, the identity information transparently "flows" into websites that require it. Our approach provides users with a consistent login experience and does not require RPs to modify their existing login forms. In addition, our approach can turn an anonymous visitor into a marketable lead with one simple click; and it could potentially decrease the sign-up form abandonment rate on RPs' websites through gradual engagement with visiting users.

3. REFERENCES

[1] R. Dhamija and L. Dusseault. The seven flaws of identity management: Usability and security challenges. *IEEE Security and Privacy*, 6:24–29, 2008.

[2] D. Florencio and C. Herley. A large-scale study of web password habits. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 657–666, New York, NY, USA, 2007. ACM.

[3] B. Freeman. Yahoo! OpenID:One Key, Many Doors. http://developer.yahoo.com/openid/openid-research-jul08.pdf, July 2008.

[4] JanRain Inc. Relying Party Stats. http://blog.janrain.com/2009/01/relying-party-stats-as-of-jan-1st-2008.html, 2009.

[5] A. Jøsang, M. A. Zomai, and S. Suriadi. Usability and privacy in identity management architectures. In *ACSW '07: Proceedings of the fifth Australasian symposium on ACSW frontiers*, pages 143–152, Darlinghurst, Australia, Australia, 2007. Australian Computer Society, Inc.

[6] E. Maler and D. Reed. The venn of identity: Options and issues in federated identity management. *IEEE Security and Privacy*, 6:16–23, 2008.

[7] C. Messina. OpenID Phishing Brainstorm. http://wiki.openid.net/OpenID_Phishing_Brainstorm, 2009.

[8] S. J. Murdoch and R. Anderson. Verified by visa and mastercard securecode: or, how not to design authentication. In *In the proceedings of Financial Cryptography and Data Security 2010*, January 2010.

[9] MyOpenID. OpenID Site Directory. http://openiddirectory.com/, 2010.

[10] OpenID Foundation. Promotes, protects and nurtures the OpenID community and technologies. http://openid.net/foundation/, 2009.

[11] OpenID Foundation. OpenID Directory. http://openiddirectory.com/, 2010.

[12] OpenID Wiki. Openid user experience. http://wiki.openid.net/browse/view=ViewFolder&param=user-experience, April 2010.

[13] D. Recordon and B. Fitzpatrick. OpenID authentication 2.0. http://openid.net/specs/openid-authentication-2_0.html, December 2007.

[14] E. Sachs. Usability Research on Federated Login. http://sites.google.com/site/oauthgoog/UXFedLogin, October 2008.

[15] S.-T. Sun, K. Hawkey, and K. Beznosov. Secure Web 2.0 content sharing beyond walled gardens. In *Proceedings of the 25th Annual Computer Security Applications Conference (ACSAC)*, pages 409–418. ACSA, IEEE Press, December 7-11 2009.

[16] The Information Card Foundation. Advance the use of Information Card. http://informationcard.net/foundation, 2009.