# Poster: OpenID$_{email}$ Enabled Browser

San-Tsai Sun (student) and Konstantin Beznosov (faculty)
University of British Columbia
Vancouver, Canada
{santsais,beznosov}@ece.ubc.ca

## 1 Introduction

With Web 2.0, the user is both a consumer and provider of Web content. However, today's Web is site centric. A user has to maintain a separated copy of identity and corresponding password for each content-hosting and service providers (CSPs), which leads to weaker passwords and/or password re-use across accounts [4].

Federated identity solutions enable cross-domain single sign-on, and remove the need for users to keep identifiers and passwords at individual CSPs. Solutions such as Liberty Alliance Project [8], Shibboleth [6], and OpenID [9] are examples of federated identity systems. However, with the exception of OpenID, current federated identity solutions require pre-established trust relationships and agreements between identity providers (IdPs) and CSPs in a federation, which limits their adoption on the Web.

OpenID is an open and user-centric Web single-sign-on protocol that does not require pre-established agreements and trust relationships between IdPs and CSPs. OpenID is user-centric in the sense that users are free to choose their own OpenID identity providers. Major CSPs (e.g., Google, Yahoo, AOL, Facebook) have already acted as OpenID IdPs, and there were over 30,000 websites supporting OpenID as a Relying Party (RP) by the end of 2008. OpenID is promising, however, for OpenID to become prevalent, there are two main issues that must be addressed.

The first issue is the usability of the OpenID identifier scheme. OpenID uses a URI as an end-user's identifier; this acts as a universal user account and is valid across all CSPs. The main advantage of using a URI as an identifier is that it can associate personal profile information and services (e.g., authentication, policy service). However, Web users perceive a URI as a "web address" instead of a personal identifier, and some OpenIDs generated by IdPs are hard to remember (e.g., https://www.google.com/accounts/o8/id?id=AItOawmY8FE from Google).

The second main issue for the OpenID protocol is with respect to phishing attacks [7]. OpenID and other similar protocols (e.g., Google AuthSub, AOL OpenAuth, Yahoo BBAuth) may habituate users to being redirected to identity provider websites for authentication. If users do not verify the authenticity of these websites before entering their credentials, phishing attacks are possible. To prevent phishing attacks, users must confirm the authenticity of an identity provider before entering their credentials. Existing research on authenticating web-sites to users include security indicators, secure bookmarks for known websites [2, 11], and automated detection and blacklisting of known phishing sites [3]. However, studies suggest that security indicators are ineffective at preventing phishing attacks [10], and blacklisting known phishing sites still suffers from high rate of false-positives and false-negatives. Even with improved security indicators, users still tend to ignore them [10].

## 2 Approach

The main idea behind our work is the metaphor of identity flows in the design of operating systems, where a user authenticates to the OS and that authenticated identity automatically flows into all processes invoked by the user. Based on this idea, our design treats a browser as an operating system and each web site a user visited resembles a process. A Web user enters her user name/password from an existing account on the Web into a browser, and that identity automatically flows into all web sites that require an authenticated identity.

Figure 1 illustrates the system architecture of our proposed solution and data flows among the main actors in the system. The main components of our proposed system are an OpenID$_{email}$ provider and an OpenID$_{email}$ enabled browser. An OpenID$_{email}$ provider is an existing email provider that augmented with both an OpenID$_{\mathrm{UA}}$ protocol and an email-to-OpenID mapping (EAUT) service [5]. OpenID$_{\mathrm{UA}}$ is an OpenID extension that we proposed to allow IdPs authenticate with a user-agent (e.g., a browser) in addition to RP web sites. As discussed, users are not accustomed to using OpenID URI as identifiers. Email ad-

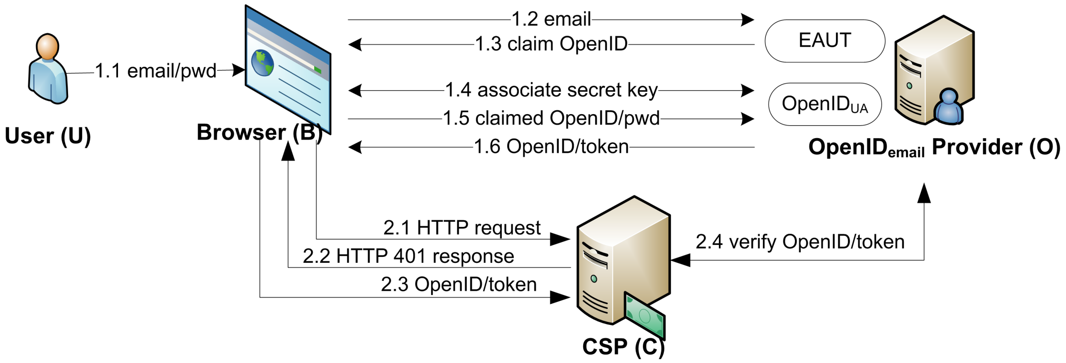

**Figure 1. Main players in the proposed Web 2.0 content sharing system.**

dresses on the other hand, have been used as user identifiers by many CSPs [1]. By combining these two services, Web users can use their email to login CSPs while remain using an OpenID identifier for identification. An OpenID$_{email}$ enabled browser is a browser extended with supports for OpenID$_{email}$ protocol and is able to flow a user's OpenID transparently to web sites that acquires it.

The following steps illustrate the sequence for a user to login into an OpenID$_{email}$ provider via an OpenID$_{email}$ enabled browser:
1.1 User **U** enters her email $e$ and password $p$ to Brower **B**.
1.2 **B** finds EAUT service **E** based on $e$ and sends $e$ to **E**.
1.3 **E** maps $e$ to an OpenID identifier $i$ and sends $i$ to **B**.
1.4 **B** discovers OpenID identity provider **O** based on $i$ and establishes a shared secret $k$ with **O**.
1.5 **B** encrypts $p$ with $k$ and sends $i$ with the encrypted password to **O**.
1.6 **O** decrypts $p$ with $k$, checks $p$, and then sends back $i$ with a token $t$ that **B** can verify using $k$.

We now provide the data flow for accessing protected content on a CSP based on our proposed approach:
2.1 User **U** uses **B** to access protected content on CSP **C**.
2.2 **C** responds with a HTTP 401 "Unauthorized" response with WWW-Authenticate authentication scheme set to "OpenID".
2.3 **B** makes a HTTP request again with $i$ and $t$.
2.4 **C** finds **O** based on $i$ and then sends $i$ and $t$ to **O** to ensure $i$ and $t$ are valid.

## 3 Conclusion

In this work, we have proposed a system to address two main issues of OpenID. The benefits of our proposed approach including: (1) **Usability**: Web users authenticate with their existing email account/password. Once logged-into an OpenID$_{email}$ enabled browser, the identity information will automatically flow into Web sites that supports OpenID for authentication. (2) **Security**: Web users do not have to enter their user name/password directly into CSPs, which reduces the chances of phishing attacks. (3) **Portability**: Web users can use an OpenID$_{email}$ enabled browser from any computer; no identity information and passwords are stored on the local machine. (4) **Privacy**: Web users' email addresses are only known to the browser; they are never revealed to CSPs.

## References

[1] B. Adida. EmID: Web authentication by email address. In *Proceedings of Web 2.0 Security and Privacy Workshop 2008*, Oakland, California, USA, 2008.

[2] R. Dhamija and J. D. Tygar. The battle against phishing: Dynamic security skins. In *SOUPS '05: Proceedings of the 2005 symposium on Usable privacy and security*, pages 77–88, New York, NY, USA, 2005. ACM.

[3] Earthlink Inc. Earthlink toolbar: scambloker for windows users, 2008.

[4] 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.

[5] D. Fuelling and W. Norris. Email Address to URL Transformation 1.0. http://eaut.org/specs/1.0/, June 2008.

[6] Internet2. Shibboleth System. http://shibboleth.internet2.edu/, 2008.

[7] B. Laurie. Openid: Phishing heaven. http://www.links.org/?p=187, January 2007.

[8] Liberty Alliance. Liberty Alliance Project. http://www.projectliberty.org/, 2002.

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

[10] S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer. The emperor's new security indicators. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, pages 51–65, Washington, DC, USA, 2007. IEEE Computer Society.

[11] M. Wu, R. C. Miller, and G. Little. Web wallet: preventing phishing attacks by revealing user intentions. In *SOUPS '06: Proceedings of the second symposium on Usable privacy and security*, pages 102–113, New York, NY, USA, 2006. ACM.