NSTIC, Privacy and Social Login

Francisco Corella
Pomcor

Karen P. Lewison
Pomcor

Revised May 27, 2011

Abstract

Social login, pioneered by Facebook Connect, is seeing rapid adoption and may soon realize one of the goals of NSTIC, viz. a drastic reduction in the number of passwords that users have to remember. But social login, as implemented today, reduces user privacy and security, gives control to the dominant social site over relying parties, and hinders competition among social sites. We suggest that NSTIC could be a catalyst in a transition to a social login that does not have such flaws, and we propose an approach to privacy-enhanced social login where an HTTP extension allows the browser to play an active role without introducing browser dependencies.

1 Motivation

The US National Strategy for Trusted Identities in Cyberspace (NSTIC) [1] has worthy privacy goals [3], which include avoiding the disclosure of unnecessary information to the relying party, the disclosure of the identity of the relying party to the identity provider, and the linking of multiple identity or attribute assertions to track the user. These goals are achievable with tools such as anonymous credentials based on zero-knowledge proofs [2].

But there is an elephant in the room that should not be ignored, viz. the recent emergence of social sites as identity providers. When a relying party delegates user authentication to a social site, it gains not only verified identity data, but also read/write access to the user's social context, including the ability to issue updates on behalf of the user that are seen by the user's friends at the site. Janrain has coined the term *social login* to refer to this combination of authentication and authorization, which was pioneered by Facebook Connect.

Social login offers a compelling value proposition to the relying party, and it is seeing rapid adoption as a Web single sign-on solution. Social login may thus realize, in the short term, one of the goals of NSTIC, viz. a drastic reduction in the number of passwords that users have to remember.

But social login, *as implemented today*, is seriously flawed along several dimensions:

Privacy. The social site is informed of every social login performed by the user, and may even be informed of the activities of the user in the Web site or application to which the user has logged in.

Security. Social login usually involves an OAuth authorization code being sent by the social site to a callback endpoint of the relying party via the user's browser. An attacker who intercepts this code can trivially impersonate the user, yet social sites do not require their relying parties to implement their callback endpoints as TLS endpoints. The user has no way to tell whether a particular relying party uses TLS or not.

Control. Social login is implemented today using OAuth, which requires prior registration of the relying party with the social site. If social login becomes a de facto user authentication standard, every Web application will have to register with the dominant social site, currently Facebook, just to be able to authenticate its users. And the dominant social site will have the powser to disable any Web application by revoking its registration.

Competition. The registration requirement puts a new social site at a competitive disadvantage because its users will not be able to perform social login, since few relying parties will be registered with the site. This competitive disadantage compounds the great competitive advantage that the "network effect" confers to the dominant social site.

A social login mechanism that does not have these flaws is very much needed, and NSTIC could be a catalyst in bringing one into existence and having it adopted.

But the need is urgent, and anonymous credentials based on zero-knowledge proofs may not yet be ready to be broadly deployed. We propose instead, at least for the short term and for the specific purpose of social login, an approach that represents an incremental improvement to the social login mechanism, based on technology that developers should be comfortable with.

2 A More Active Role for the Browser

The approach that we are proposing uses two ingredients: one-time key pairs, and an extension of HTTP that allows the browser to take a more active role without creating browser dependencies.

Many existing identity and social login protocols, including Windows Live, SAML Browser SSO Profile, Shibboleth, OpenID and OAuth, use a double-redirection redirection where the relying party redirects the browser to the identity provider, which authenticates the user and redirects the browser back to the relying party. The extension of HTTP that we have in mind would define an enhanced double-redirection mechanism that woudd be explicitly supported by HTTP, rather than being a "trick" implemented via the 302 status, which was intended for a very different purpose, or via form submission by JavaScript code.

The following high-level sketch of the proposed approach assumes that a suitable HTTP extension is available without attempting to define it. The proposed social login mechanism comprises the following steps, where all connections are protected by TLS.

Step 1. The relying party initiates the enhanced double-redirection by redirecting the browser to the social-site. The relying party sends the browser a social login request that specifies the user attributes and the scope of access to the user's account that the relying party wants to obtain from the social site. The relying party also sends the browser a callback URL and a one-time public key, which is the public key component of a one-time key pair generated by the relying party.

Step 2. The browser retains the callback URL rather than forwarding it to the social site.[1] The browser generates another one-time key pair and retains the private key component. The browser sends to the social site the social login request, the relying party's one-time public key, and the browser's own one-time public key.

Step 3. The social site silently authenticates the user using, for example, a TLS client certficate submitted by the browser, or an authentication cookie that refers to an existing social-site login session. If user authentication is by username and password, the user must be logged in ahead of time; this requirement is a countermeasure against phishing

[1] The hiding of the callback URL by the browser can be optional. The relying party can be allowed to specify, in step 1, that the browser should forward the callback URL to the social site.

attacks. After authenticating the user, the social site signs a one-time certificate binding the requested attributes to the browser's one-time public key, and a one-time certificate binding a grant of access to the user's account to the relying party's one-time public key. Then the social site redirects the browser back to the relying party, sending the two one-time certificates to the browser, as well as a user identifier. The identifier is a secret high-entropy random string that uniquely identifies the user within the social site. The social site does not specify the callback URL, which it does not know; this will be supplied by the browser.

Step 4. The browser asks the user for permission to proceed with the social login and to provide the requested attributes and access to the user's account to the relying party. The relying party is identified to the user by data extracted from its TLS certificate, which the browser obtained as it established the connection that carried the HTTP response that initiated the enhanced double redirection.

Step 5. The browser computes a user identifier specific to the relying party by concatenating the identifier provided by the social site and the domain name of the relying party, and applying a crypotographic hash function to the concatenation. The browser completes the enhanced double-redirection by sending an HTTP request to the callback URL that it retained in step 2, using the one-time certificate with the user's attributes as TLS client certificate, and using the corresponding one-time private key, which it also retained in step 2, in the TLS handshake. The payload of the request carries the user identifier specific to the relying party and the one-time certificate with the access grant.

Step 6. The relying party accesses the user's account at the social site as needed over TLS connections, using the one-time certificate with the access grant as TLS certificate.

3 Properties of the Proposed Approach

The proposed approach provides strong security and has the following privacy properties:

1. The social site does not know what relying party is performing the social login. This property has multiple benefits:

   (a) The social site cannot track the user's activity on the Web.
   (b) The user is free to choose any relying party, without requiring approval of that party by the site.
   (c) The social site does not have the power to disable a relying party by removing its registration.
   (d) New social sites can offer social login to any relying party, just like the dominant social site.

2. The user attributes will typically include personally identifiable information such as a name and a photograph, or information unique to the user such as a pseudonym and an avatar. Such information allows colluding relying parties to track the user. However, if the attributes include no unique information, then colluding relying parties cannot track the user because the attributes are bound to different public keys for different relying parties.

The one-time public keys are known to the social site and the relying party. They can therefore be used to identify the user involved in a social login if the social site colludes with the relying party, and to track the user if it colludes with multiple relying parties. However, since the social site is online, the same identification and tracking can be achieved simply by timing correlation.

4 Conclusion

Social login reduces the need for creating paswords, and has compelling advantages for relying parties. But, as implemented today, it reduces user privacy and security, gives control to the dominant social site over relying parties, and hinders competition among social sites. NSTIC could be a catalyst in creating a social login mechanism that does not have such flaws. We have proposed an approach to social login that could be used to that purpose. The approach is based on one-time key pairs and an enhanced double-redirection mechanism explicitly supported by an HTTP extension that allows the browser to play a more active role, without introducing browser dependencies.

5 Acknowledgements

Kim Cameron pointed out to us that one-time public keys can be used to identify and track the user if the social site colludes with relying parties.

This is a motivation for using zero-knowledge proofs in cases where the identity provider does not have to be online.

Craig Wittenberg pointed out to us that that the user's identifier can be used by colluding relying parties to track the user, and suggested that the browser could make the identifier specific to the relying party.

References

[1] Jeremy Grant. National Strategy for Trusted Identities in Cyberspace, April 6, 2011. Presentation at NIST IDtrust 2011, available at http://middleware.internet2.edu/idtrust/2011/slides/03-national-strategy-trusted-identities-cyberspace-nstic-grant.pptx.

[2] Brian A. LaMacchia. New results using anonymous credentials: Constrained delegation and revocation, April 2011. Presentation at NIST IDtrust 2011, available at http://middleware.internet2.edu/idtrust/2011/slides/02-privacy-lamacchia.pptx.

[3] Howard A. Schmidt. The National Strategy for Trusted Identities in Cyberspace and Your Privacy, April 26, 2011. White House blog post, available at http://www.whitehouse.gov/blog/2011/04/26/national-strategy-trusted-identities-cyberspace-and-your-privacy.