



SOCIAL-NETWORKS CONNECT SERVICES

Moo Nam Ko, Gorrell P. Cheek, and Mohamed Shehab, *University of North Carolina at Charlotte*
Ravi Sandhu, *University of Texas at San Antonio*

New services such as Facebook Platform, Google Friend Connect, and MySpaceID let third-party sites develop social applications without having to build their own social network. These social-networks connect services increase access to and enrich user data in the Social Web, although they also present several security and privacy challenges.

Social-networking websites let users build social connections with family, friends, and coworkers. Users can also build profiles for storing and sharing various types of content with others, including photos, videos, and messages. Updating user profiles with interesting content is a form of self-expression that increases interaction in such sites. To encourage this interaction and provide richer content, social-networking sites expose their networks to Web services in the form of online application programming interfaces. These APIs allow third-party developers to interface with the social-networking site, access information and media posted with user profiles, and build social applications that aggregate, process, and create content based on users' interests.

Social-networking sites provide numerous application services that can mash up user-profile data with third-party data. In addition, third-party sites can rapidly distribute their services via social-networking sites to keep

in touch with users while they're on these sites. Moreover, users can enjoy various applications with content from numerous third-party sites: users access social-networking sites, where they maintain their profiles; third-party sites retrieve these profiles, enrich the content, and return them to the social-networking sites for consumption by the user and, possibly, friends. For example, Facebook users can share music with friends, create playlists, and get concert alerts on their profile page by installing the third-party music application iLike (www.ilike.com).

Major social-networking sites have begun launching *social-networks connect services* such as Facebook Platform, Google Friend Connect, and MySpaceID that further break down the garden walls of social-networking sites. These SNCSs let third-party sites develop social applications and extend their services without having to either host or build their own social network. This extension allows third-party sites to leverage the social-networking site's features.

For example, third-party sites can exploit the authentication services provided by a social-networking site so that users need not create another username and password to access the third-party site; instead, users can draw on their social-network credentials and established profile. Users can also access third-party sites that leverage social-network user-profile content. The third-party sites retrieve users' profiles from the social-networking site to create an enhanced experience. In this way, they can increase membership by providing more interesting content from a variety of sources in a seamless manner.

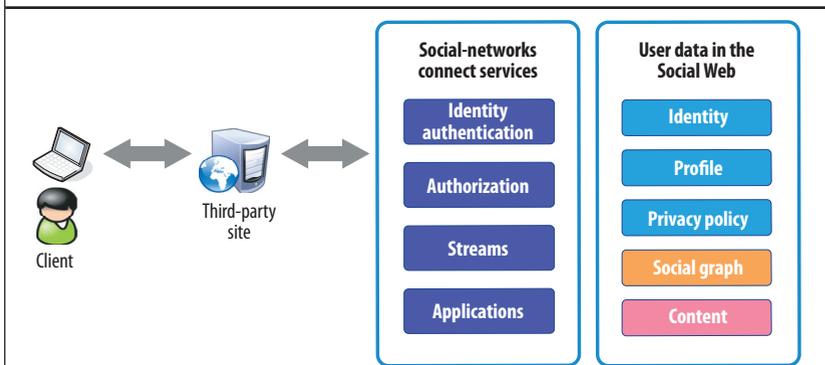


Figure 1. Social-networks connect services framework.

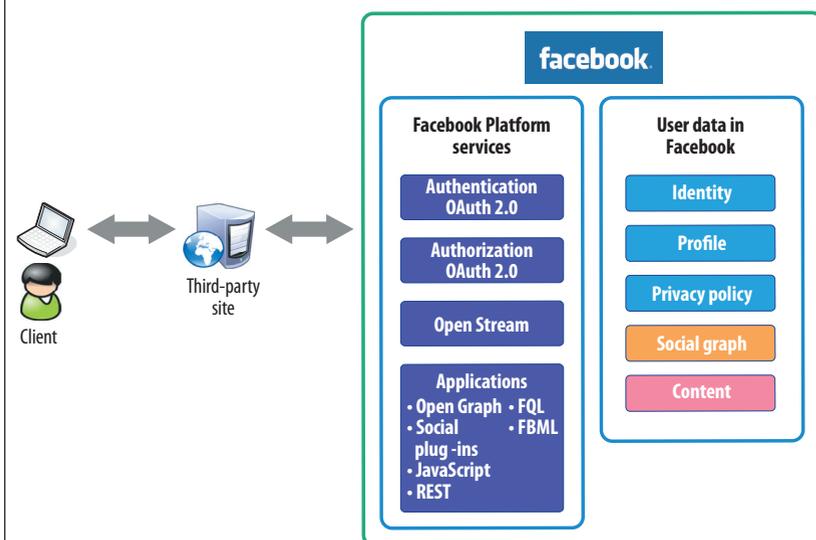


Figure 2. Facebook Platform services.

SOCIAL-NETWORKS CONNECT SERVICES

User data is composed of three types of information. *Identity data* describes who I am in the Social Web, including my identity, profile information, and privacy policy. *Social-graph data* represents who I know in the Social Web, including my friendship connections with descriptions such as family, coworker, colleague, and so on. *Content data* represents what I have in the Social Web, including my messages, photos, videos, and all other data objects created through various Social Web activities.

For social-networking sites to be able to share user Social Web data with third-party sites, a secure and reliable SNCS framework is required. As Figure 1 shows, this framework consists of a collection of four categories of APIs that allow third-party sites to interface with the social-networking site:

- *Identity authentication* proves users' identity; users can authenticate using their existing accounts from various identity providers to include the social-networking site.

- *Authorization* governs access to user data in the Social Web based on pre-defined authorization access rights; the authorization API lets third-party sites create new content and extract existing content from users' Social Web data.
- *Streams* let third-party sites publish to users' activity streams and vice versa.
- *Applications* let third-party sites develop rich social features in the form of applications and thereby extend the Social Web.

The implementation of these APIs can vary widely with different protocols and technologies.

Facebook Platform

Facebook Platform became generally available in December 2008 under the Facebook Connect brand. (Facebook announced in April 2010 that it's eliminating the Facebook Connect brand as part of the launch of the Open Graph API; it's not clear at this point whether the company will replace it or call it something else.) Since then, more than 80,000 third-party sites have implemented it. Facebook Platform lets third-party sites integrate with Facebook and send information both ways to create more engaging and richer social experiences on the Web. As Figure 2 shows, Facebook Platform allows

users to import their identity, profile, privacy policy, social graph, and content from Facebook to third-party sites.

Authentication is by far the most used Facebook Platform component. This API enables third-party sites to leverage Facebook as an identity provider. For example, Digg.com, a social news website where users can share content, doesn't require new members to register and create a profile. Instead, as Figure 3 shows, they can use their existing Facebook profile to authenticate (steps 1 and 2); once authenticated, new users can extend their social graph to the third-party site and invite friends—or link to them, if they're already members—to join them on Digg.com (step 3).

Facebook Platform leverages OAuth 2.0 for authentication and authorization. OAuth 2.0 is a simplified, improved version of the Open Authorization standard (<http://oauth.net>) that lets third-party sites obtain authorization tokens from Facebook. First, a user of the third-party site authenticates using Facebook as an identity provider. Next, Facebook issues a token that lets the third-party site

access the user's basic profile information including name, picture, gender, and Friend List. The third-party site can request extended permissions depending on the specific application requirements—for example, to access the user's Wall, the space on the user's Facebook profile where friends can post messages. The third-party site can also apply Facebook users' privacy policy settings. For example, if Alice doesn't allow Bob to access her content on Facebook, then Bob wouldn't be allowed to access Alice's content on the third-party site.

The Open Stream API lets third-party sites read and write to users' activity streams—for example, short messages on new events. This API supports multiple-stream publishing methods as well as the Atom feed standard. Third-party sites can read content from users' activity streams in addition to publishing to their activity streams. For example, Facebook users can share their activities on third-party sites with their friends on Facebook through the News Feed feature.

Facebook provides a series of APIs to assist developers in creating social applications that interface with third-party sites (<http://developers.facebook.com/docs>). The primary API is Open Graph, which lets third-party applications read and write content objects—photos, friends, and so on—and the connections among them in Facebook's social graph. The API is simple in that it allows access to content objects via URLs. For example, with proper authorization, <http://graph.facebook.com/FBUserID/friends> will provide access to FBUserID's friends. Facebook also provides support for the Representational State Transfer (REST) API. However, future enhancements will mainly focus on the Open Graph API.

Facebook's social plug-ins also enable traditional third-party sites to expand into the Social Web with minimal HTML. The Like button lets users share pages and send content back to their Facebook profile. The Recommendations plug-in allows users to suggest or recommend content on the third-party site. The Activity Feed plug-in shows Facebook users what their friends like. Facebook offers various other social plug-ins including Comments, Facepile, Login, and Live Stream.

In addition, Facebook has a JavaScript software development kit consisting of classes and methods that can be

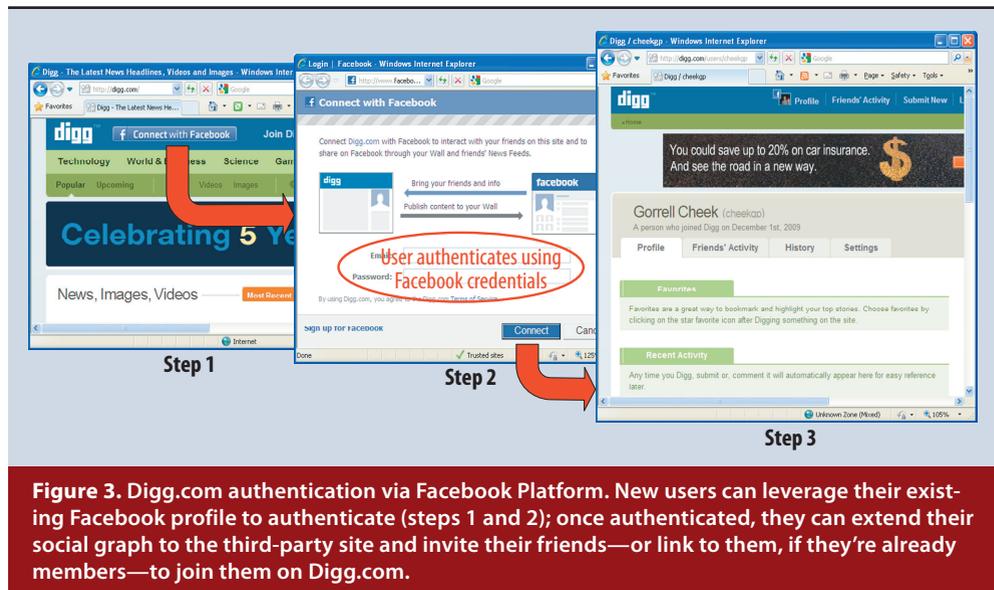


Figure 3. Digg.com authentication via Facebook Platform. New users can leverage their existing Facebook profile to authenticate (steps 1 and 2); once authenticated, they can extend their social graph to the third-party site and invite their friends—or link to them, if they're already members—to join them on Digg.com.

used to integrate third-party sites using Facebook Platform along with numerous tools for developers to build social applications. Facebook Markup Language (FBML) is a proprietary variant of HTML, and Facebook Query Language (FQL) provides a quick and easy mechanism to query Facebook user data without using API methods.

Google Friend Connect

Released at approximately the same time as Facebook Platform, Google Friend Connect also makes it easy to share profile, social-graph, and content data with third-party sites (www.google.com/friendconnect). As Figure 4 shows, its decentralized approach to integrating social and nonsocial websites relies on open standards such as OpenID (<http://openid.net>), OAuth, and OpenSocial (www.opensocial.org).

OpenID lets users authenticate to third-party sites using credentials issued by a supported OpenID identity provider such as Google or Yahoo. This API prevents users from having to go through the new-member registration process. After initial authentication, users can select their existing profile from a profile provider site like Plaxo. They can then import their social graph from a social-networking site such as Orkut to share their activities in the third-party site. For example, if a user posts a message on a board in a third-party site, the message is only visible to friends in the selected social graph.

OAuth provides granular authorization control of user content in the Social Web based on privacy policies. Users can share content hosted on a third-party site without having to provide a username and password to the requesting site. OAuth issues authorization to a specific site for a specific content object for a specified time. For example, assume site A houses photos on behalf of a user and site B requires access to these photos. Site B requests access

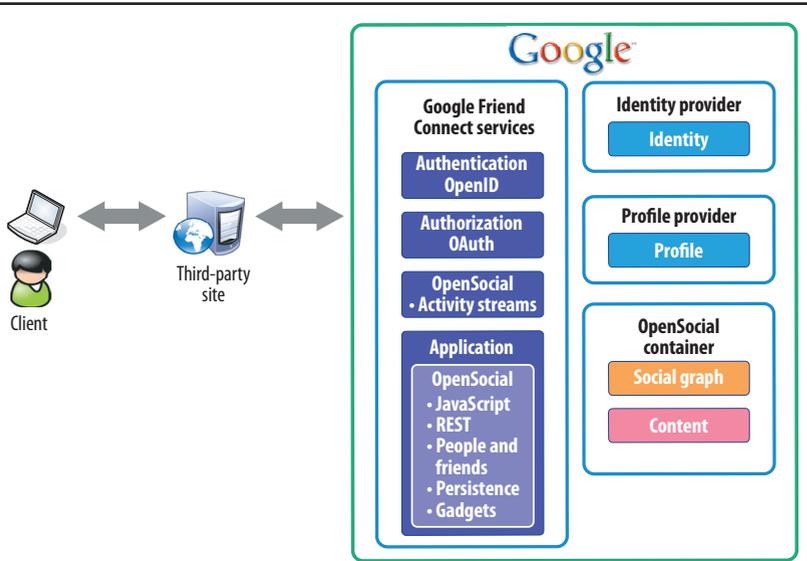


Figure 4. Google Friend Connect services.

from the user, who then authorizes the access request by authenticating to site A and authorizing site B's access request. Site A will then issue an authorization token for the photos to site B. Site B never gains access to the user's site A credentials.

OpenSocial is that social applications can potentially reach more users if they integrate with multiple container sites, and, theoretically, users will have access to more social applications. OpenSocial's client-side API leverages standard Web development technologies such as JavaScript, HTML,

and Ajax (Asynchronous JavaScript and XML). It offers server-side development via the RESTful data APIs, which expose three primary sets of data: people and friends—who I am and who I know (social graph); my activities; and *persistence*—the ability to read and write data with your friends (content).

Many third-party sites that adopt Google Friend Connect insert client-side HTML/JavaScript applets, or *gadgets*, into their pages. In this way, the sites can quickly add social features by simply integrating a few snippets of code. Different social gadgets, such as a rating gadget—which lets users rate a movie—easily transform the third-party site from being part of the Web to being part of the Social Web. The downside of this approach is that gadgets can't access social graphs or content; developers must use JavaScript or RESTful data APIs to overcome this limitation.

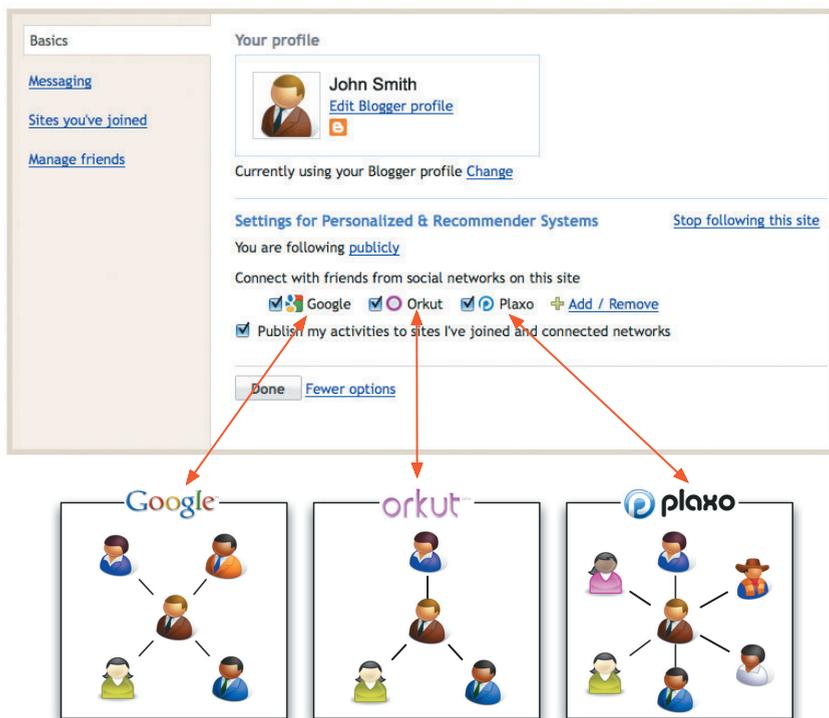


Figure 5. Google Friend Connect blog site profile page. The OpenSocial API reads and writes to users' activity streams on Google Friend Connect sites, enabling users to share what they're doing and to see what their friends are up to.

MySpaceID

MySpace has adopted Google's

Table 1. Social-networks connect services comparison.

API category	Facebook Platform	Google Friend Connect	MySpace ID
Authentication/authorization	Single service provider with 400 million users Open standard: OAuth 2.0	Multiple service providers Open standards: OpenID, OAuth	Single service provider with 113 million users Open standards: OpenID, OAuth
Streams	Full support Proprietary	Full support Open standard: OpenSocial	Full support Open standard: OpenSocial
Applications	Full suite of APIs and tools Proprietary	Full suite of APIs and tools Open standard: OpenSocial	Full suite of APIs and tools Open standard: OpenSocial

open-standards approach to Social Web development, likewise relying on standards such as OpenID, OAuth, and OpenSocial to share profile, social-graph, and content data with third-party sites (<http://developer.myspace.com/myspaceid>), as Figure 6 shows. MySpaceID lets users log in to third-party sites with their MySpace credentials as well as to share their profiles and social graphs with such sites; it also provides support for activity streams. OpenSocial and MySpace's own RESTful APIs are the primary resources for creating third-party social applications. Because MySpace is an OpenSocial container, MySpace user data is easily accessible to any third-party site using the standard.

SNCS comparison

Facebook Platform, Google Friend Connect, and MySpaceID enable third-party site developers to integrate with the Social Web without having to build their own social network. Using these SNCSs, users can leverage their existing identity, profile, social graph, and content on multiple social-networking sites. All three SNCSs provide integration APIs and tools.

Table 1 highlights the similarities and differences among these SNCSs. One major difference is that Google Friend Connect uses decentralized identity, profile, and social-graph providers while Facebook and MySpace leverage their own social-network platform for user data. The advantage of the decentralized approach is that users can select from among the best of all possible worlds to customize their presence on the Social Web; on the other hand, users must maintain their social data in multiple locations, which increases administration costs. Using a single-service provider cuts down on this administrative overhead but also limits users' capabilities and choices to that of the one provider.

In addition, Google and MySpace follow an open-standards approach while Facebook has traditionally taken a proprietary approach, though recently it

has adopted OAuth. Standards-based social applications should reach more users because there are so many standards-based service providers. However, these users are also spread across various service providers. Servicing 400 million users with one provider, as Facebook does, creates significant mass in the overall Social Web. Not surprisingly, third-party sites want access to Facebook's membership because it represents an extremely large user base.

OPEN RESEARCH CHALLENGES

The Social Web is growing exponentially due to SNCSs, but with this growth come several challenges, most pertaining to security and privacy.

Identity mapping

Users often have multiple identities scattered across various social-networking and third-party sites. For new users of third-party sites, SNCSs eliminate the need for a cumbersome registration process. They also enable users to link an existing account on a third-party site to one on a social-networking site, thereby providing a seamless Social Web experience. For example, Facebook compares the hash values of a user's e-mail address at Facebook with

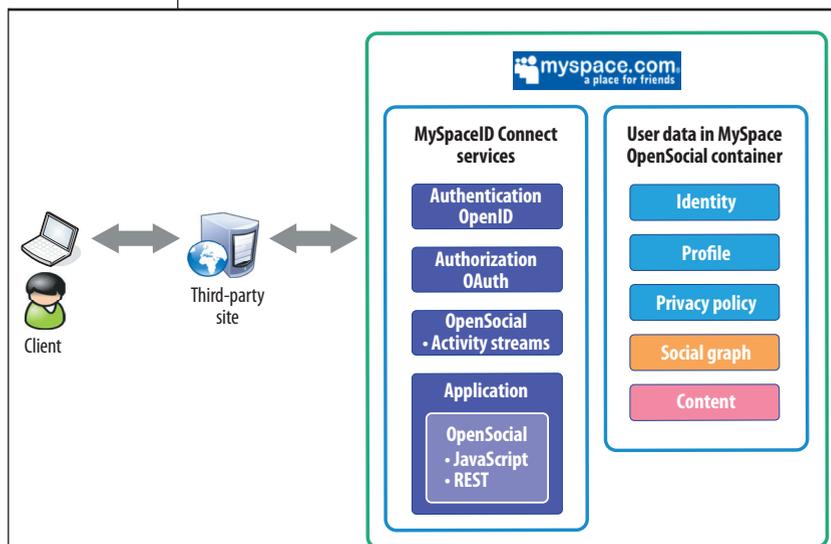


Figure 6. MySpaceID services.

that at a third-party site and, if they match, links them together. It also uses this approach to connect a user's Facebook friends with third-party site friends. However, this won't work if a user's e-mail address on Facebook differs from that on the third-party site. Moreover, if the user's friends have different e-mail addresses, linking their accounts is almost impossible.

Mapping friends' accounts is especially important in protecting user privacy on the Social Web. Unfortunately, current identity-mapping methods like that used by Facebook have many shortcomings. One ideal solution is that all users have a global unique identity offered by an OpenID service provider. However, OpenID isn't well known and hasn't been fully adopted.

Researchers have investigated attribute-based identity mapping. However, G. Alan Wang and colleagues revealed that incomplete records would significantly increase the comparison algorithm's error rate.¹ This is a common limitation of many identity-matching techniques that use only personal attributes. Jennifer Xu and colleagues showed that combining social features with personal features could improve criminal identity matching.² However, this approach may not be reliable in today's Social Web as the quality of user attributes in profiles is low due to deception, errors, or missing attributes.

Mapping multiple identities issued by different identity providers is another issue in SNSs. Google Friend Connect provides authentication services using multiple identity providers. If a user logs in to a third-party site using one identity provider and later logs in to the same site using a different identity provider (assuming the user has identities with both), the third-party site will treat this user as two separate identities and the user will have two profiles, two activity streams, two sets of friends, and so on. Given the fact that most users easily forget their usernames and passwords, adding an additional dimension—requiring them to select their identity provider—will add a further burden.

User data portability

Social Web users often maintain multiple profiles and social graphs on different social-networking sites. For example, a user may have a Facebook account to keep in touch with family and friends and a LinkedIn account for professional contacts. Managing these disparate profiles and social graphs can be a chore. One possible solution is for users to maintain a global profile and social graph in one place. They could then create subprofiles tailored to a specific audience—for example, family and friends, colleagues, or other members of a virtual community built around a particular interest—and set appropriate privileges on these subprofiles depending on the context.

There are challenges with this approach: identifying a single profile repository, resolving duplicate profile attributes, and name-space mapping across different social

graphs. Michael Sutterer, Olaf Droegehorn, and Klaus David proposed an ontology of user subprofiles based on a specific context.³ Their approach is primarily geared toward personalized services on mobile communications platforms but could be applied to the Social Web. For example, a user could maintain a global profile with multiple subprofiles that are made available depending on the type of site being visited.

Common enhanced privacy policy framework

Social-networking and third-party sites have their own access-control mechanisms based on privacy policies that use different and often incompatible policy language. In addition, current SNS implementations limit the ability to set and enforce a policy on user data. Facebook Platform extends users' privacy controls to social applications hosted on third-party sites, but only if the application developer fully implements these capabilities. The approach Google Friend Connect and MySpaceID use to manage the privacy of user profiles and content is even more immature.

Several researchers have investigated novel access-control schemes for the Social Web. The Lockr system lets users manage a single social graph with access-control lists; only those who have a social relationship described in an ACL can access the shared content.⁴ Mohammad Mannan and Paul C. van Oorschot proposed an access-control model based on existing "circles of trust" in instant messaging networks.⁵ And San-Tsai Sun, Kirstie Hawkey, and Konstantin Beznosov proposed an e-mail-based system in which users leverage their OpenID identities and content owners use their contact lists to specify OpenID providers' access policies.⁶ The problem with these approaches is that access control is based on one e-mail contact list, IM network, or social-networking site and thus isn't applicable across multiple social graphs.

Applying a generic privacy policy across all social-networking and third-party sites presents a major challenge. Just as OAuth has propelled authentication and authorization, a common enhanced privacy policy framework based on open standards must be designed and deployed. However, building access controls that use compatible policy language and deliver a consistent and satisfactory user experience will require extensive collaboration within the community.

Cascaded authorization

Developers can combine data or functionality from two or more third-party sites to provide a new aggregated service. Third-party sites frequently rely on other third-party sites to provide services, which requires the sharing of user data from one third-party site to another. Under current authentication and authorization approaches, users must consent each time a third-party site accesses their data. This approach is both cumbersome and time-consuming.

What's needed is a new authorization mechanism that would obtain a user's consent once for a specific content object and cascade it to all third-party sites that use that content object as part of a mashup service.

The state of the art is OAuth, which enables third-party sites to share tokens to delegate access rights to other third-party sites. The challenge raised by this approach is that it requires users to trust third-party sites to make authorization decisions on their behalf. OAuth for Recursive Delegation is a proposed extension of OAuth that lets a client (or third-party site) delegate the authorization it has received from a user to another client.⁷ However, this is an emerging standard with no concrete implementations.

Data integrity in social plug-ins

SNCSs let people share not only site URLs, events, and photos with others but their opinions as well. For example, users can share comments about a video they posted on YouTube with friends on a social-networking site. Facebook's social plug-ins are widely deployed. Several third-party sites have adopted the thumbs-up Like button to enable users to easily post their opinions on those sites. The "liked" Web object is referenced and shared using the object URL, which makes it easy for a user's friends to access the shared object. However, although social plug-ins enable the seamless sharing of user opinions, they don't guarantee data integrity. For example, what happens if the content of a "liked" object changes? Adequate data integrity controls for social plug-ins remain an open research challenge.

Many third-party sites have adopted social-networks connect services to extend their presence in the Social Web. Integrating these third-party sites with SNCSs creates a more feature-rich online social community and promises to break down the garden walls of social-networking sites. However, many challenges come with this growth, and the social-networking community must collaborate to design and deploy secure services that both protect privacy and deliver a satisfactory user experience. 

References

1. G.A. Wang et al., "Automatically Detecting Criminal Identity Deception: An Adaptive Detection Algorithm," *IEEE Trans. Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 36, no. 5, 2006, pp. 988-999.
2. J. Xu et al., "Complex Problem Solving: Identity Matching Based on Social Contextual Information," *J. Assoc. Information Systems*, vol. 8, no. 10, 2007, pp. 525-545.
3. M. Sutterer, O. Droegehorn, and K. David, "Making a Case for Situation-Dependent User Profiles in Context-Aware Environments," *Proc. 2007 Workshop Middleware for Next-Generation Converged Networks and Applications (MNCNA 07)*, ACM Press, 2007, pp. 1-6.
4. A. Tootoonchian et al., "Lockr: Social Access Control for Web 2.0," *Proc. 1st Workshop Online Social Networks (WOSP 08)*, ACM Press, 2008, pp. 43-48.
5. M. Mannan and P.C. van Oorschot, "Privacy-Enhanced Sharing of Personal Content on the Web," *Proc. 17th Int'l Conf. World Wide Web (WWW 08)*, ACM Press, 2008, pp. 487-496.
6. S.-T. Sun, K. Hawkey, and K. Beznosov, "Secure Web 2.0 Content Sharing beyond Walled Gardens," *Proc. 2009 Ann. Computer Security Applications Conf. (ACSAC 09)*, IEEE CS Press, 2009, pp. 409-418.
7. B. Vrancken and Z. Zeltsan, "Using OAuth for Recursive Delegation," v01, Internet Eng. Task Force draft, Feb. 2010; <http://tools.ietf.org/html/draft-vrancken-oauth-redelegation-01>.

Moo Nam Ko is a PhD student in the Department of Software and Information Systems, College of Computing and Informatics, University of North Carolina at Charlotte. His research interests include user-centric identity and privacy management, secure content-sharing frameworks, and social-application development and methodologies. He is a graduate student member of IEEE and the ACM. Contact him at mnko@uncc.edu.

Gorrell P. Cheek is a PhD student in the Department of Software and Information Systems, College of Computing and Informatics, University of North Carolina at Charlotte. His research interests include information security and privacy, with a focus on access-control methodologies. He is currently researching machine-learning techniques for governing access within large online social networks. Cheek is a member of the IEEE Computer Society and the ACM. Contact him at gcheek@uncc.edu.

Mohamed Shehab is an assistant professor in the Department of Software and Information Systems, College of Computing and Informatics, University of North Carolina at Charlotte. His research interests lie in network and information security, especially in the design and implementation of distributed access-control protocols to cope with the requirements of emerging distributed social networks, Web services, and peer-to-peer environments. Shehab received a PhD in computer engineering from Purdue University. He is a member of the IEEE Computer Society and the ACM. Contact him at mshehab@uncc.edu.

Ravi Sandhu is founder and executive director of the Institute for Cyber Security, holds the Lutchter Brown Endowed Chair in Cyber Security, and is a professor in the Computer Science Department of the College of Science, with joint appointments in the College of Business and College of Engineering, at the University of Texas at San Antonio. He is also cofounder and chief scientist of TriCipher, an identity solution provider for businesses based in Los Gatos, Calif. His research focuses on high-impact research, practice, and education in cyber security. Sandhu received a PhD in computer science from Rutgers University. He is a Fellow of IEEE, the ACM, and the AAAS. Contact him at ravi.sandhu@utsa.edu.

 Selected CS articles and columns are available for free at <http://ComputingNow.computer.org>.