Privacy Delegate: a browser-based tool for privacy self-management in social networks

Miguel A. Monjas, Jose M. del Alamo, Juan-Carlos Yelmo, Jonas Hogberg

***Abstract*—This position paper presents a user-centric schema for self-managed privacy that enables users to use a dashboard to find out which user information a social network provider has shared with and to rule the way such a sharing procedure is done, according to some privacy policies set by the user. The proposal is implemented as a browser plug-in and works on the OAuth protocol.**

***Index Terms*—Browsers, Identity Management Systems, Privacy, Social network services**

I. MOTIVATION

Online social networks on the Web are becoming one of the most popular services in the Internet. It refers to services such as Facebook, LinkedIn or MySpace that focus on building and reflecting social relations between people. They enable users to socialize in a virtual fashion with their family, friends, workmates, and any other kind of social groups.

An online social network service (SNS) essentially consists of a digital representation of each user (usually a profile including personal information), his/her social links, and a variety of additional identity-based services e.g. private messaging, discussion forums or communities, personal event management, and so on and so forth. One of the most common ways users socialize in this new networks is by sharing personal information (identity attributes such as name, birth date, gender, friends, hobbies, favorite reading, music, football team and so on) and personal multimedia content generated by him/her (photos, videos, etc.), also known as "social information". On the other hand, online social networks are evolving from supporting their users in sharing personal information within the provider domain towards enabling personal information exchange with third parties.

As a result, for example, Facebook Connect ([1]) and MySpace Developer Platform ([2]) both provide Application Program Interfaces (APIs) for third party developers to gather users' personal information, social graph, etc.

However, although users may have become comfortable about sharing more information, more openly and with more people than ever before [3], they are also increasingly concerned about retaining control over their personal information [4]. In addition, legislation usually forces SNS providers (to different extents, depending on the jurisdiction) to handle user data in accordance with the law.

In response, SNS providers are increasingly offering new solutions to improve the privacy of their users. In order to offer a better service, many have created extensive sets of privacy controls that allow users to toggle between different levels of confidentiality for their personal information. However, the resulting heterogeneity is a huge disadvantage for users who wish to manage and control their personal information actively. As concluded from a thorough analysis of 45 social networking sites [5], "privacy in social networks is dysfunctional in that there is a significant variation in sites' privacy controls, data collection requirements, and legal privacy policies".

In summary, users currently lack a simple mechanism to verify what personal information is available on different SNS's, how it is used and how they can modify, update or delete it. Since users are not offered a comprehensive privacy management service, privacy-awareness is expected to increase in the future. In this report, we introduce a solution that aims to address some of these problems by enabling users to manage their personal data privacy in SNS's in a simple and efficient way.

II. PRIVACY IN SOCIAL NETWORKS

Privacy in SNS's has become a social issue and a hot research topic [5] [6] [7]. However, most of the related work focuses on information sharing among users of the same SNS. With the introduction of new APIs that allow third parties to gain access to users' personal and social information new problems have come about for the SNS providers. They have tried to extend existing tools to control what information is available to installed applications. Some SNS's have also introduced APIs that require explicit user consent for each

This work has been partially supported by CDTI, Ministry of Science and Innovation of Spain, as part of the SEGUR@ project (https://www.cenitsegura.es/), within the CENIT program, with reference CENIT-2007 2004.

M.A. Monjas is a senior researcher with the Technology and Innovation unit at the Ericsson R&D Center in Madrid, Spain (e-mail: miguel-angel.monjas@ericsson.com).

J.M. del Alamo is an associate professor at the Department of Telematics System [IST] Engineering belonging to the Universidad Politécnica de Madrid (UPM), Spain (e-mail: jmdela@dit.upm.es).

J.C. Yelmo is a full professor at the Department of Telematics System [IST] Engineering belonging to the Universidad Politécnica de Madrid (UPM), Spain (e-mail: jcyelmo@dit.upm.es).

J. Hogberg is a senior systems engineer with the Development Unit Core and IMS at the Ericsson R&D Center in Madrid, Spain (e-mail:jonas.k.o.hogberg@ericsson.com).

transaction related to users' personal resources hosted by the SNS.

Recently, Felt and Evans carried out a study on information sharing with third party applications in SNS's [9] and addressed the privacy risks associated. The authors propose a new API for personal information sharing based on a *privacy-by-proxy* design, which restricts access to user data by hiding inappropriate user data from unauthorized viewers and anonymizing users' social graphs. This privacy-by-proxy solution is aimed at being integrated with SNS APIs, which is not always possible. Additionally, it does not solve the problem that users face when their information is scattered around different SNS's.

Recently, a series of open standards collectively known as the Open Stack has emerged to cope with the heterogeneity of personal and social information sharing on the Web. The Open Stack refers to a set of open technologies that work together to make it easier for developers to manage access to users' information across the Web: OpenID [11], Extensible Resource Descriptor Sequence (XRDS) [12], Portable Contacts [13], OpenSocial [14] and OAuth [8]. OpenSocial is a set of APIs that details methods for accessing information about people, their friends, and their personal information. Third party developers can create applications that take advantage of users' personal resources hosted by SNSs that have implemented the OpenSocial APIs.

OpenSocial relies on OAuth to manage access to users' social information. OAuth introduces a third role to the traditional client-server authentication/authorization model: the resource owner. In the OAuth model, the client (which is not the resource owner, but is acting on his/her behalf) requests access to resources controlled by the resource owner, but hosted by a container i.e. the SNS. OAuth allows the SNS to verify the identity of the client making the request as well as ensuring that the resource owner has authorized the transaction. Following OAuth procedures, which are based on user-agent redirections, resource owners are able to authorize third-party access to their resources without sharing their credentials.

OAuth uses tokens generated by the container instead of the user's credentials in their requests for Protected Resources. The process uses two token types: request tokens and access tokens. Request tokens are used by the consumer to ask the user to authorize access to the Protected Resources. The user-authorized request token is then exchanged for an access token. Access tokens are used by the consumer to access the Protected Resources on behalf of the user.

OAuth authentication is the name the OAuth specification gives to the process in which a user grants access to their protected resources at a given container to a consumer without sharing their credentials at said container with the consumer. The process comprises three consecutive steps, which are usually triggered when the user carries out a transaction in the consumer (depicted in Figure 1, step A):

1 The consumer obtains an unauthorized request token from the container (Figure 1, step B).

2 Through user interaction, s/he authorizes the request token. First, a consumer provides the user with an unauthorized request token that the user agent redirects to the container (Figure 1, step C). Then, the container begins an authentication dialogue with the user (Figure 1, step D) which, if successful, implicitly authorizes the request token. Finally, the container provides the user with an authorized request token (Figure 1, step E), which he/she redirects back to the consumer.

3 Finally, the consumer exchanges with the container the authorized request token for an access token (Figure 1, step F) in order to subsequently access the information.

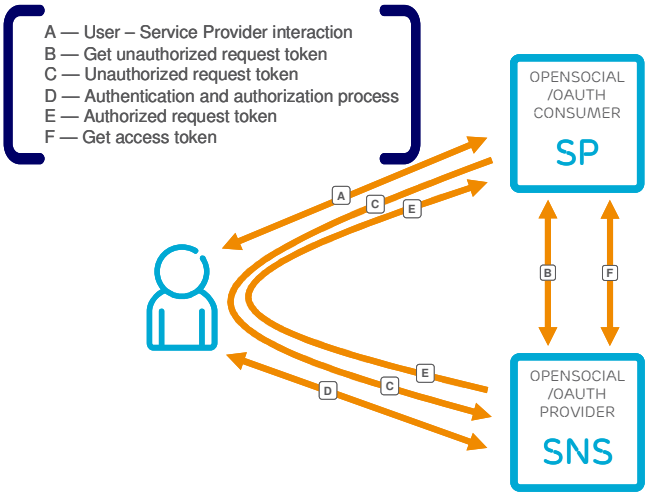

Fig. 1. OAuth basic authorization procedure

Within the OpenSocial/OAuth model, privacy is handled on-line (live) by the user, who grants one-time authorization each time a consumer tries to access personal information stored in a specific OpenSocial container. Even if the execution of steps A and F needn't the user to be online with neither the consumer or with the container, steps C, D and E only can happen if the user is online with the consumer.

OAuth initially did not provide any method for scoping the access rights granted to a Consumer. A Consumer either has access to Protected Resources or hasn't. Many scenarios will, however, require greater granularity of access rights. In this case, users might explicitly set further ad-hoc privacy controls in each social network they belong to, as long as the social network allows it. Such privacy settings may restrict third-party applications to access the users' data.

III. PRIVACY DELEGATE

A. *Self-Service Privacy*

We define Self-Service Privacy as the quality that allows individuals to control their sensitive personal digital information explicitly and consciously in a safe and easy way. A Self-Service Privacy realization must allow its users to

manage their distributed personal information from a central control point. This central point does not have to store any personal information but just discover, retrieve and allow the user to manage it as a kind of personal *digital identity dashboard*.

Additionally, the central point must allow users to govern their information completely, both tracking previous access to it and controlling its future use and release, thus performing as a *personal privacy dashboard*. To achieve this, collaboration and coordination with the providers storing the personal information is a must. We envisage that in the near term governments will force providers to declare the personal information they store and to provide standard-based mechanisms to interact with it. As a matter of fact, this process has already started in some European countries introducing legislative principles such as 'privacy by design' and 'personal information accountability' [16]. Current technologies, such as those of the Open Stack, partially support these features.

B. *Self-Service Privacy in social networks*

Ericsson has been very active in the Liberty Alliance and Kantara Initiative fora. One of the main functional areas covered by such Liberty Alliance, especially when devising ID-WSF was privacy management. Ericsson developed the concept of Privacy Self-Service [13][17] to address the problem of privacy handling across different service providers in a network-centric identity management architecture as for the Liberty Alliance ID-WSF specifications [15].

In order to cope with the privacy requirements described in the motivation section, Ericsson propose the introduction of a new entity in the OpenSocial/OAuth environment, namely the Privacy Delegate (PD). This new entity shall provide functionalities similar to those of the Privacy Self-Service. Using this Privacy Delegate, users are able to:

1 Retrieve a global view (snapshot) of their identity resources scattered in different nodes of the identity network i.e. know what identity resources are stored, where they are distributed and their specific values. Users are also allowed to manage (modify and cancel) the values shown.
2 See the history of use of the identity resources i.e. what entities have requested them, when, the outcome of the process, etc.
3 Govern the future use and release of the identity resources by setting privacy preferences that are distributed to entities hosting user's identity information.

The PD leverages on OpenSocial/OAuth in order to provide users with both a static and a dynamic view of their social resources in a given social network and a convenient way to centrally set and delegate the enforcement of their privacy preferences for the governance of this information. The static view allows users to know what resources are stored within a social network (OpenSocial container) i.e. a snapshot of their social resources. The dynamic view allows users to know how this information has been requested by and released to different service providers (OpenSocial consumers).

However, the main functionality provided by the PD is the setting of privacy preferences for the access to their social resources. It works on behalf of the user, intercepting the messages exchanged between Consumers and Containers, so that users' privacy preferences can be enforced without repeatedly needing online (live) user authorizations. Additionally, the exchange of authorizations to access social resources can be accounted, so that future audits are possible. In that sense, the PD replicates the Privacy Self-Service concept, as it enables users to fully manage the information exchanged by social network providers with other external

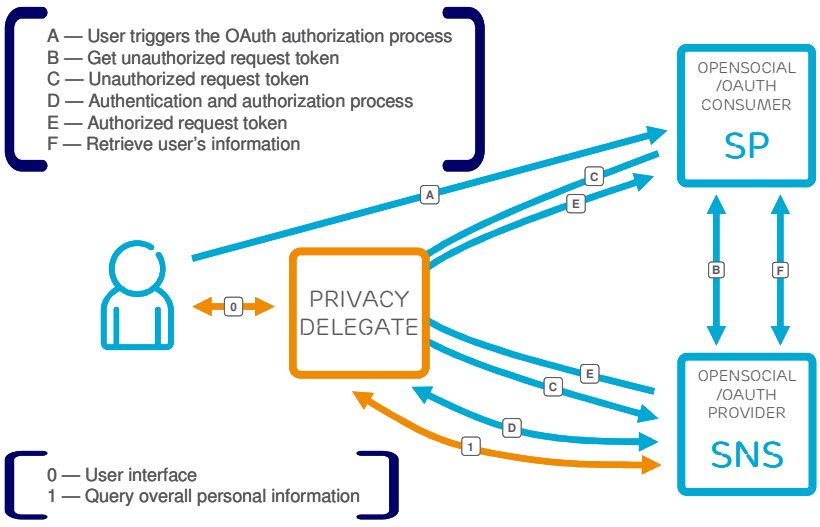

Fig. 2. Privacy Delegate solution

entities. The PD may store user credentials in different social network sites in order to allow deferred interaction.

The solution implemented by the Privacy Delegate is not intended to replace the standard mechanism foreseen by the OpenSocial/OAuth specifications, but to extend and complement it in order to provide users with a single control point for identity and privacy management in social networks.

We have implemented the Privacy Delegate as a plug-in within the user's browser to make it possible to catch the consumer initial request. This approach does not impact any of the OAuth entities. Provided that the browser handles the OAuth authorization procedure, it is the natural place to store and enforce user's decisions on sharing information with third parties and applying them to further requests.

Figure 2 introduces a high-level view of an SNS (Open Social container) sharing its users' personal and social information in accordance with the OAuth protocol. It also shows the information flows involved. There is a communication flow between the user and the PD through a Graphical User Interface (GUI) (Figure 2, flow 0). Using the GUI the PD shows information to users and, in response, users take certain actions and decisions, which are sent back to the PD in order to drive the aforementioned functionality.

Flows labeled from A to F correspond to standard messages as defined by the OAuth/OpenSocial specification. (A) A user triggers the process by requesting some action from a service provider (called consumer in OAuth parlance). To complete the transaction the consumer requires some personal information that is stored at a given SNS (called container in OAuth parlance). In order to gain access to that information the consumer requires (B) a request token (unauthorized at that moment) from the container. (C) The consumer sends a message back to the user containing the unauthorized request token and redirects their user-agent to the container. When the container receives the message, it begins the user authentication process (D) which, if successful, also involves the authorization of the token. Then, the container sends (E) the authorized token through the user back to the consumer. Eventually, (F) the consumer uses the authorized token to retrieve an access token that will allow subsequent access to the social information stored in the SNS.

The core functionality provided by the Privacy Delegate is privacy management. Users may set privacy preferences to govern the use and release of their personal information, which is stored in an SNS. The PD will show different options to users and allows them to configure different parameters such as the conditions under which the data can be released. When finalized, the PD stores the resulting privacy policies. From that point on, the PD enforces these privacy preferences.

Users may express their privacy preferences by different means. First they can choose one out of several pre-defined privacy policies and associate it to a social resource. These pre-defined privacy policies are described in natural language so that non-technically skilled users can understand them. This natural language description is mapped to a specific policy implementation described in a privacy policy expression language. These policies are hierarchical so that it is easier for users to compare among them and choose the one that better suits their needs. The approach benefits from the simplicity and usability of the model because users do not have to deal with the policy details.

Users are also allowed to define each detail of the privacy policy. Although this approach provides great flexibility in the description of users preferences it poses some risks for the usability: just advanced users understand (and probably want to know) the meaning of the policy. This is offered as an advanced option.

Once the privacy policies are set (there must be a "deny all" default policy), the enforcement of such policies may start. The PD must enforce the policies thus deciding whether the service provider receives the authorized token that allows it to retrieve the information.

When the service provider begins the authorization process as for OAuth it sends an HTTP message to the user's user agent redirecting him/her to the SNS for authorization. The PD intercepts this message and extracts the information relevant for policy enforcement, i.e. the applicant (consumer identifier). As OAuth did not define initially how to retrieve other information, namely the protected social information and the operation the consumer wants to be authorized, we defined two new parameters that all OAuth authorization messages should include to pass on such information. If these parameters are not included in the authorization request the PD might assume that the consumer tries to query, modify and delete all the user's social information stored in the SNS, and will enforce the privacy preferences accordingly.

Once the privacy policies have been enforced, and the result of this enforcement has been allowed, the PD goes on with the OAuth protocol. Thus the PD forwards the original request token for authorization to the SNS. Then the SNS starts the authentication procedure with the user through his user-agent and the PD intercepts the message again. It can be assumed that the credentials needed for the authentication are available in the user's browser (they could have been stored when retrieving the static view of the user's profile), so explicit permission by the user is not needed unless the user has stated that he must be asked for consent. The PD sends the authentication response and the SNS sends back the authorized request token, which the PD forwards to the service provider. The service provider can go on with the regular OAuth protocol by exchanging the authorized request token for an access token at the SNS and then gaining access to the protected social information and operation needed.

If the result of the policy enforcement process at the PD results in a refusal, then the PD returns a message back to the service provider with information on the details of the refusal.

IV. CONCLUSION

Privacy Management has been always one of the main

components of Identity Management. However, although there have been countless efforts to improve the way end users handle the privacy of their personal data, there is no tool to manage personal data beyond some vertical applications tightly tied to the service or application where user data is hosted or processed.

In that sense, Ericsson has proposed different approaches to enable users to find out which personal information have been "leaked" and the way it must be processed by any other entity. A first approach relied on network-centric identity management technologies such as those of Liberty Alliance. Keeping as requirements its main functionalities and acknowledging that user-centric technologies such as those of the so-called Open Stack had gained preeminence, a new alternative was devised. The Privacy Delegate enables user to find out which personal information has been shared by service network providers and to rule the way such a sharing procedure must be carried out. The user browser is the natural location for this tool, as the browser is the client that indirectly handles the sharing of information, taking an active part in the authorization procedure.

Future developments should address a seamless integration with user authentication or secure synchronization of privacy preferences among different user browsers thus providing a consistent user experience. In addition, other approaches to identity and privacy should be considered. For example, how to govern the increasing and uncontrolled, but potentially harmful, user-related information delivered by third parties out of the user control e.g. in social networks where some users gossip about others.

REFERENCES

[1] Facebook Connect, http://developers.facebook.com/connect.php
[2] MySpace Developer Platform, http://developer.myspace.com/
[3] Kirkpatrick, M.: "Facebook's Zuckerberg Says the Age of Privacy is Over", ReadWriteWeb, Jan 9 2010. Available at http://www.readwriteweb.com/archives/facebooks_zuckerberg_says_the_age_of_privacy_is_ov.php
[4] "Privacy 2.0: Give a little, take a little", The Economist, Jan 28 2010. Available at http://www.economist.com/specialreports/displaystory.cfm?story_id=15350984 (Premiun content)
[5] Bonneau, J., Preibusch, S.: "The Privacy Jungle: On the Market for Data Protection in Social Networks". In: 8th WS on the Economics of Information Security (2009).
[6] Gross, R., Acquisti, A., Heinz, H.J.: "Information Revelation and Privacy in Online Social Networks". In: 2005 ACM Workshop on Privacy in the Electronic Society, pp. 71 – 80, ACM Press, Ney York (2005)
[7] Aimeur, E., Gambs, S., Ho, A.: "UPP: User Privacy Policy for Social Networking Sites". In: 4th International Conference on Internet and Web Applications and Services, pp. 267–272, IEEE CS, Los Alamitos (2009).
[8] Hammer-Lahav, E. (Ed.): "The OAuth 1.0 Protocol", IETF RFC 5849, April 2010. http://tools.ietf.org/html/rfc5849
[9] Felt, A., Evans, D.: "Privacy Protection for Social Networking Platforms". In: Web 2.0 Security and Privacy 2008, Oakland (2008).
[10] Del Alamo, J.M. et al.: "Self-Service Privacy: User-Centric Privacy for Network-Centric Identity". In: 4th IFIP International Conference on Trust Management, Springer (2010)
[11] OpenID Foundation Website, http://openid.net
[12] Wachob, G. et al (Eds.): "Extensible Resource Identifier (XRI) Resolution Version 2.0, Committee Draft 03", 28 February 2008.
[13] Portable Contacts Website, http://portablecontacts.net
[14] OpenSocial Website, http://www.opensocial.org/
[15] Liberty Alliance specifications, http://www.projectliberty.org/resource_center/specifications/liberty_alliance_complete_specifications_zip_package_05_may_2009/
[16] Article 29 of the Data Protection Working Party, The Future of Privacy - Joint contribution to the Consultation of the European Commission on the legal framework for the fundamental right to protection of personal data, 02356/09/EN, 01 Dec 2009.
[17] Del Álamo, J.M., Fernández, A.M., Trapero, R., Yelmo, J.C., Monjas, M.A.: "A Privacy-Considerate Framework for Identity Management in Mobile Services". In Springer Mobile Networks and Applications (in print).