Making Privacy a Fundamental Component of Web Resources

Thomas Duebendorfer (Google Switzerland GmbH), Christoph Renner (Google Switzerland GmbH/ETH Zurich), Tyrone Grandison (IBM), Michael Maximilien (IBM), Mark Weitzel (IBM)

We present a social network inspired and access control list based sharing model for web resources. We have specified it as an extension for OpenSocial 1.0 and implemented a proof of concept in Orkut as well as a mobile social photo sharing application using it. The paper explains important design decisions and how the model can be leveraged to make privacy a core component and enabler for sharing resources on the web and beyond using capabilities of mobile devices.

1. INTRODUCTION

Based on prior work that focused on making privacy a fundamental building block for social networks [1-3], we propose APIs and infrastructure enhancements to the Web. These modifications ensure that the disclosure intent of the creator of every Web resource is known prior to those resources being rendered by a Web client. Our upgrades also take into consideration backward-compatibility with current Web protocols and components.

Though *privacy* is still a concept-in-flux, with many interpretations, unspoken meanings and facets, we support the following views on privacy. The first is that privacy is concerned with how much control I have over the use and disclosure of my data. The second is that privacy is highly context-dependent, subjective, dynamic and temporal, i.e. I may consider a piece of information sensitive if it is to be used for purpose X at one point in time and may consider it not sensitive for the same purpose at a different point in time or in a different situation. The third is that privacy controls often built on top of security controls but not vice versa. There exist unified frameworks that support both strong security and privacy at the data level [4].

In addition to the recent public interest in breaches of what was thought to be private data on social networking websites, privacy is critical to the social networking industry because it empowers the user base and maintains and supports the trust fabric, which is a core assumption for these systems. This fabric allows these companies to execute their business strategy, despite it being potentially damaging to their users - who are the providers of their business value. At the heart of the issue, there is a conflict of interest between social network users and social network platform owners. The platform owners want the right to unconditionally use all the information given or generated by the users; while those users may desire specific data use under specific conditions. This perceived bifurcation raises the opportunity to create privacy-preserving sharing technologies that meet both the business owner needs and the system user requirements.

Thanks to the popularity of social networks there exist several hundred million online identities [6] and therefore the Internet no longer has to be an anonymous place. These social systems can be used for controlling who gets access to which uploaded information.

In Section 2, we propose a simple access control model that provides the basis for enabling user privacy - the specification of who has access to my data and what operations they can perform. We allow to configure the visibility [1,2] of shared data items. In Section 3, we explicit the applicability of our access control model from profile data to group information to activity streams to media items and beyond. In Section 4, we purport that the assumption that every single data item has an associated access control list (ACL) is one that can be extrapolated from the social networking paradigm to the Web. By example of our sample implementation [3], we present easy solutions to the issues of ACL bloat and seamless staged integration into the existing infrastructure. It is important to note that our ACL approach differs from traditional ACLs because the closed system assumption of traditional ACLs must be removed. In this context, subjects and targets may be email addresses, automated computing agents, etc. Also, the allowable actions could be Web-oriented actions like "share via X", where X could be a twitter name, email address, facebook id, etc. Thus, our proposal extends the contemporary access control models.

2. DESIGN

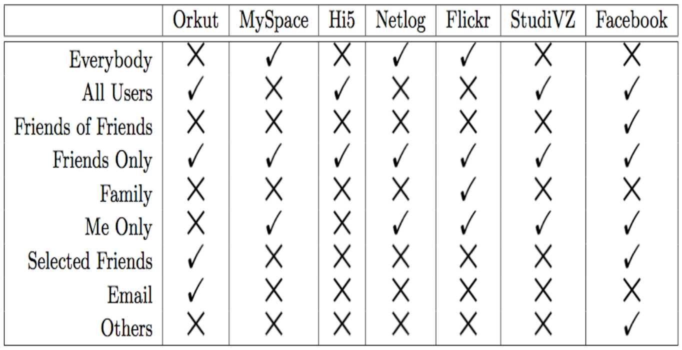
Giving the user control and transparency over his/her data is the basis for enabling privacy on the web. While designing our access control model, we aimed to create a solution that is of immediate benefit to as many users as possible and we required that it supports automation (e.g. by offering APIs). We've also found that a few social networking platforms like Facebook and MySpace have started to provide proprietary solutions for controlling access to specific shared data types like media items within their respective network. However, they are are proprietary and not compatible with each other.

With more than 36 social networks currently supporting OpenSocial [5], having a combined user base of more than 600 million users [6], we decided to design and propose generic access control lists as an extension to the OpenSocial version 1.0 specification. The access control lists (ACLs), which are attached to a shared data item, can be created, modified and deleted by the owner of that data item.

Three fundamental questions for any access control schema are:

1. Which entities can a data item be shared with?
2. How are these entities specified?
3. How is access granted or to rephrase in our context: what's the sharing model?

We got inspired by examining a various entities present in today's social networks and found that support for sharing and types of entities known are rather inconsistent as the following table shows.



	Orkut	MySpace	Hi5	Netlog	Flickr	StudiVZ	Facebook
Everybody	✗	✓	✗	✓	✓	✗	✗
All Users	✓	✗	✓	✗	✗	✓	✓
Friends of Friends	✗	✗	✗	✗	✗	✗	✓
Friends Only	✓	✓	✓	✓	✓	✓	✓
Family	✗	✗	✗	✗	✓	✗	✗
Me Only	✗	✓	✗	✓	✓	✓	✓
Selected Friends	✓	✗	✗	✗	✗	✗	✓
Email	✓	✗	✗	✗	✗	✗	✗
Others	✗	✗	✗	✗	✗	✗	✓

Figure 1: Supported entities for sharing media items found in web user interfaces of social networks (March 2010).

We found that the following list supports the most common entities well. We restricted the types to just four to keep it simple, while still allowing for extensibility. We use the BNF notation present in the OpenSocial specification 1.0 because our proposal extends that specification.

```
ACL-Entry-Type = "GROUP" / "USER"
/ "EXTERNAL_CONTACT" / "CUSTOM"
```

ACL-Entry-Type	**description**
GROUP	A group as defined in OpenSocial 1.0.
USER	A single social network service user.
EXTERNAL_CONTACT	An external contact, which is not a user of the social network service.
CUSTOM	This value must only be used if none of the above is appropriate. It indicates an

	extension proprietary to the OpenSocial container providing it.

Table 1: Entities in our access control model

For GROUP, we predefined the following ids.

```
Group-ID = Object-Id / "@self" /
"@friends" / "@all" / "@everybody"
/ "@family"
```

Group-ID	description
@self	When used in the context of a user, this group contains only that user.
@friends	Contains all the user’s friends. The distance of the friends from the owner is set in ACL attribute `networkDistance`.
@all	Contains all the users in that OpenSocial container.
@everybody	Contains all Internet users.
@family	Contains all the users which belong to a user’s family (a special group defined by the user).

Table 2: Predefined group entities in our access control model

EXTERNAL_CONTACT represents entities in the form of Accessor-Types, which are not present in the social network user base. This allows for sharing through communication channels like phone or email that leave the scope of the social network.

```
Accessor-Type = "MAILTO" / "PHONE"
/ Custom-Accessor-Type
```

Accessor-Type	description
MAILTO	An email address. Shared via email e.g. by sending a link.
PHONE	A phone number. Shared with someone who is specified by his phone number e.g. by sending a text message.
Custom-Accessor-Type	A proprietary extension for additional external contracts proprietary to the OpenSocial container that provides it. These Custom-Access-Types must use a prefix starting with a lowercase letter.

Table 3: Predefined external contact entities in our access control model

The next fundamental question to address is the sharing model. Should it be additive (i.e. whitelist), subtractive (i.e. blacklist) or a combination of both? Given that a combination of both makes interpretation really hard for the user but also for client applications, especially if only the container (i.e. server) has complete knowledge of group memberships, we decided to only support the additive model. The item is always shared implicitly with the owner if there is no ACL. Any other entity that needs access must be added explicity to the ACL. This is easy to understand by the end user and simple to implement. There's one caveat. Unlike traditional access control models in a closed system, where users are typically managed by a single adminitrator, we allow for entities with rather weak identities like dynamically growing and shrinking groups or external contacts like email addresses, which could even be mailing lists.

An ACL-Entry contains also accessor rights besides the just discussed Acl-Entry-Type. Possible accessor rights on the web resource are “GET”, “POST”, “PUT”, “DELETE”. They match the available operations in OpenSocial's RESTful API.

Putting it all together, an actual access control list is defined as a list of ACL-Entry elements, which are attached to a web resource in a social network. Additionally, we added the optional attribute "numberOfPeople" to individual ACL-Entry elements as well as to the access control list. This lets the OpenSocial container quantify how many people in total currently have access to this resource (remember that groups are dynamic in social networks), which gives a rough sense of how private a sharing might be. Finally, we also included an optional "fields" attributed to list which fields of a web resource (think of a person profile in a social network) this access control list applies to.

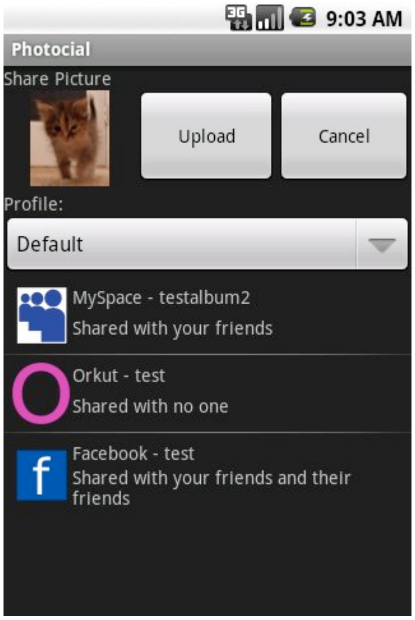
A complete formal specification with some more features not described here can be found in our proposal sent to the OpenSocial community on Feb 28, 2010, which is described in full length in the appendix of [3] and which is currently under discussion for inclusion in OpenSocial version 1.1.

We hope to have found a good balance between specification simplicity, generic design and expressability of the access control scheme.

3. APPLICABILITY OF ACLs

Our proposal for access control lists for OpenSocial was designed to support a wide range of web resources present in social networks. We worked out two formal proposals (both can be found in the appendix of [3]) to extend OpenSocial version 1.0 to use them for sharing activities, media items and albums, which contain a number of media items. Furthermore, we also thought of how to use them for controlling and exposing access to profile information of social network users, where some networks requires some groups of fields to have the same access level (e.g. not shared at all or only friends can see both fields age and sex). This is the main reason why we introduced he field list attribute in the access control list.

To get a sense of the implementation complexity, we implemented our access control lists in Orkut for media items and albums. The integration was straight forward and was done as part of a Master's Thesis at Google. In order to use our access control lists, we implemented a mobile application for the Android platform to seamlessly share photos between and amongst several social networks..



MySpace - testalbum2
Shared with your friends

Orkut - test
Shared with no one

Facebook - test
Shared with your friends and their friends

Figure 2: Our Android app that uses our Opensocial access control model, which we implemented in Orkut.

4. SHARING WEB RESOURCES

While this paper has discussed access control lists with a focus on OpenSocial so far, we posit that our access control model can also be applied to web resources outside of social networks. The Web no longer has to be an anonymous place. Hundreds of millions of users [6] have a profile on one or more social networking sites. Additionally, the OpenId [7] authentication standard is gaining popularity, which supports a multitude of identification mechanisms. These already existing verifiable entities and the fact that the web has become social in many ways, could be used to apply access control for sharing web resources. Using a RESTful API to set, modify and delete access control lists as used in our proposal comes very natural for managing access to web resources.

Furthermore, we imagine that certain trusted mobile and client applications will get the capability to enforce access control locally on the device or computer. A local contact manager on a mobile phone could e.g. push photos uploaded to a personal blog in a social network (after asking the user for permission) to all people in the friends category of the local contact manager and share them by email or MMS, whatever contact information is present, and hence allow for enabling access control way beyond a single social network.

5. CONCLUSION

The web is at a tipping point to turn social in many ways. A core thing that is missing, is a sharing model for web resources that is generic enough to be applicable to today's social networks and to a multitude of data types like media items, activities, albums, profiles and arbitrary web resources. We proposed an access control model, which we specified as an extension to OpenSocial version 1.0 and implemented for media items and albums in Orkut as a proof of concept. Our proposal of a generic access control list could serve as the basis for controlling access to web resources not just in social networks but anywhere on the web, where users can be authenticated when needed, e.g. through OpenId, in order to grant access. We envision that our access control model can even span beyond the web by leveraging the capabilities of mobile phones to share information consistent with the user's preferences across different communication channels. To complement our access control model, agreeing on common user interface design elements for sharing would help to make the use of access control a seemless experience and to prevent unintended sharing configurations.

6. REFERENCES

1. Maximilien, E.M., Grandison, T., Sun, T., Richardson, D., Guo, S., Liu, K. "Enabling Privacy As a Fundamental Construct for Social Networks". 2009 IEEE International Conference on Social Computing (SocialCom-09). Vancouver, Canada. Aug 2009. URL: http://www.almaden.ibm.com/cs/projects/iis/hdb/Publications/papers/SPOSN2009.pdf
2. Liu, K., Terzi, E. "A Framework for Computing the Privacy Scores of Users in Online Social Networks". IEEE International Conference on Data Mining (ICDM 2009). Miami, Florida. USA. December 6-9, 2009. URL: http://cs-people.bu.edu/evimaria/papers/pr.pdf
3. Renner, C. „Privacy in Social Networks“. Masters Thesis. 2010. ETH Zurich.
   URL: ftp://ftp.tik.ee.ethz.ch/pub/students/2009-HS/MA-2009-11.pdf
4. Agrawal, R., Grandison, T., Bird, P., Logan, S., Rjaibi, W., Kiernan, G. "Extending Relational Database Systems to Automatically Enforce Privacy Policies", 21st International Conference on Data Engineering, 2005. URL: http://www.almaden.ibm.com/software/projects/iis/hdb/Publications/papers/fgac_icde05.pdf
5. OpenSocial. Opensocial website. http://www.opensocial.org, 2010.
6. Blog „Opensocial is 1 and reach is 600 million users at 20 sites“. http://blogs.sun.com/socialsite/entry/opensocial_is_1.
7. OpenID Foundation website. http://openid.net/