Enforcing Access Control in Social Network Sites

Filipe Beato, Markulf Kohlweiss, and Karel Wouters*

Katholieke Universiteit Leuven
Dept. Electrical Engineering - ESAT/SCD/IBBT-COSIC
Kasteelpark Arenberg 10, Leuven-Heverlee (Belgium)
(firstname.lastname)@esat.kuleuven.be

Abstract. Confidentiality and data handling are important issues for social network users. Ideally, access control enforcement should not depend on the social networking provider but should be under the control of the user. In this paper, we propose a practical, SNS platform-independent solution, for social network users to control their data. We develop concepts that are general enough to describe access control restrictions for different SNS platforms. Our architecture uses encryption to enforce access control for users' private information based on their privacy preferences. We have implemented our model as a Firefox extension.

1 Introduction

Social network sites (SNS), such as Facebook[1], MySpace[2], Hi5[3], LinkedIn[4], are a popular and useful tool for people to share information [2, 1, 8]. At the same time, SNS are dangerous due to the possibly unwanted disclosure of information. This happens because it is hard to control who accesses which information. SNS providers offer some mechanisms to enforce access control, but this model requires users to rely on the provider, who may not always be trustworthy. We propose a model and a solution to address this problem, by providing users with a tool to control their own data by means of encryption.

While even privacy aware SNS users want to share selected information with a selected audience groups, they might want to make the information visible only to a limited audience by creating a white-list. This is referred to as audience segregation [7].

While sharing information, social network users face a privacy and usability paradox. On the one hand, the access control mechanisms provided by social

* All authors have been supported in part by the Concerted Research Action (GOA) Ambiorics 2005/11 of the Flemish Government, by the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy), and in part by the European Commission through the ICT program under the following contract: ICT-216483 PRIMELIFE.

[1] http://www.facebook.com
[2] http://www.myspace.com
[3] http://www.hi5.com
[4] http://www.linkedin.com

networking sites are often extremely coarse; e.g. often all of their contacts are categorised as friends and share the same access rights. On the other hand, with the growth of the presented privacy configuration options, there is the potential for misconfiguration, and outright conflict between different configuration settings. Moreover, the social network provider has still access to all of their personal data.

In this paper, we present our research work on a mechanism and a prototype that allows not only for the definition of access control rules for audience segregation, but also for the support of their enforcement. To support the definition of access control rules we develop concepts that are general enough to describe access control rights for a variety of different social networking sites. We also investigate different means for enforcing access control using encryption techniques.

We implemented a Firefox extension that provides the enforcement mechanism. The extension knows about the users' access control preferences and enforces it using encryption techniques. For the future we plan to integrate the extension with different social networking sites to automatically obtain a list of a user's connections and their grouping into different audiences. This should reduce the configuration and key management effort that users need to invest before being able to use our system.

Outline. In Section 2 we present related work in the area of privacy-enhancing technologies for SNS. We describe our attacker model in Section 3. In Section 4 we describe our system and give implementation details. We conclude by discussing remaining problems and propose future work for a more usable solution.

2 Related Work

The need for selective access control has been identified within previous works on Social Network Sites (SNS) [2, 1, 8, 11]. These works try to raise awareness for the need for privacy in social networks. This might lead to social network providers improving their service and privacy enforcement mechanisms to take the social network users' privacy needs into consideration.

Famous social network sites, such as Facebook and MySpace, already present mechanisms to enforce users' adjustable privacy preferences, by labelling data to limit access control as private, public or visible by group of friends. This means have been introduced, in Facebook, due to some privacy activist groups complains, on the News feed options. Thus, in this case, Facebook by having access to all the information that each user posts, may utilise it in their business model by offering targeted advertisement.

There has been some research and work done in the area of protecting private information within social network sites. The project Lockr[5] was initiated by a group of researchers from University of Toronto [3], and offers social network

[5] http://www.lockr.org/

users' access control of their sharing data by hiding and mapping the selected information into a third-party storage. As an example, images, could be hidden in a storage server like Picasa[6]. The main concern with the Lockr extension is the need to rely on a trusted third party storage for the hidden information.

The above authors do not consider the enforcement of selective access control through cryptographic means, rely heavily of the authentication system of third-party storage systems.

This issue was, e.g. raised by the NOYB [9] that encrypts personal information using a pseudo-random substitution cipher. The cipher replaces a personal data entry with a pseudo randomly selected substitution taken from a public dictionary. However, their approach works only for encrypting personal data from a relatively small domain, and does not allow to encrypt free text entries such as frequently found in a social network.

Another approach is flyByNight [12] by Matthew Lucas and Nikita Borisov. In their paper, they present a Facebook application to protect private data by storing it in Facebook in encrypted form. Their application is SNS dependent and relies on the SNS own servers for key management. The decryption algorithm is implemented in JavaScript and retrieved from Facebook. Therefore, their scheme while being browser independent and portable (it supports the use of arbitrary internet terminals as long as they support JavaScript), it is not secure against active attacks by the social network provider, Facebook. In contrast, our prototype application tries to rely primarily on the user side and has no dependencies from any SNS, as it is entirely client-side dependent.

The biggest gap of the above schemes is the lack of selective access control. While NOYB assumes a shared secret key that is known to the social network user circle of friends and friends of friends, flyByNight requires the social network user to select one by one each users to which a message should be encrypted. However, our approach and the NOYB approach can be seen as complementary. To extend NOYB with selective access control secret keys used by NOYB to randomise substitutions could be encrypted and thus distributed to different audiences using our approach.

3 Attacker Model

Shared information and connections among users in the social network has direct influence on the users' privacy. Social network users are exposed to the following attackers: the social network provider, the users from the social network, users that are registed in the social network but do not belong to their circle of friends. These attackers target different privacy flaws.

In terms of external users, users that are not connected to the social network, mechanisms can be implemented by SNS providers. These mechanisms protect social network users' information from non-authenticated users.

Users' may want to shield their data from certain users, that are directly or not connected to them.

[6] http://picasa.google.com

However, providers are the strongest attacker. Social network providers have access to all users' private information, and thus they can use it for several purposes. Massive targeting advertisement and behaviour analysis by using data mining techniques are just examples. The SNS providers can also share social network users' information with large companies or research groups, or provide access for governments for surveillance purposes.

In our implementation, we mainly target the scenarios where the social network provider is the attacker. However, by enforcing access control by means of encryption we protect social network users from other possible attackers. By allowing users to define on-client preferences and encrypt the content posted into the server controlled by the provider, users will assure that their personal data will be readable only for a selective audience. This will also offer a more fine-grained selection and control relative to content shared with other users within the social network.

4 Our Solution

Users need to be able to have control over their own data, and specify who can access it, preferably without putting trust into third parties, such as the SNS server providers. Our solution allows users to restrict access rights for a selective audience, and enforce access control on the target data using encryption.

4.1 Solution Model

Social networking sites represent a large virtual community that due to all social network users' connections represent a large directed graph, assuming that friendship may not be mutual. Each social network user profile contain information about his data and connections. Therefore, to manage the role-based access control, in order to allow the social network user control over his own private data, we propose a tree-like structure of the user profile node. Thus, we categorise the social network user profile in two types classes:

1. *Connections classes* which classify the social network user's connections, such as Friends, Family or Co-Workers. These classes represent groups and can be divided into sub-groups;
2. *Content classes* which classify social network user's content data. These content data can also be divided into sub-classes such as data related to hobbies, family, or work.

Formally, in order to define access control, we considered that each class represents a set. Thus, a set A is a sub-class of class B if $A \subset B$. In this way users connections and content form a partially ordered set (lattice).

4.2 Access Rights

The mapping between content and connection classes defines the access control rights. The class structure allows easy propagation of rights, without overloading the social network user. When a new information item is introduced in a content class, all members that belong to a connection class and that have access rights to the content will have access to that information item. Similarly, when a new connection is added to a connection class, it will have access to all information items to which his peers also have access. Due to the fact, that the access control enforcement for the social network user information is done by the user itself, using the prototype application on the client side, the social network provider will not learn who has access rights to what.

This model allows the user to control access to his information in a very fine-grained way. While technically skilled users might find this interesting, less computer-savvy people need to be provided with some default classes of connections and data, and preferably also a default mapping, in which a privacy level might be specified. In this way, we try to create a good balance between usability and confidentiality.

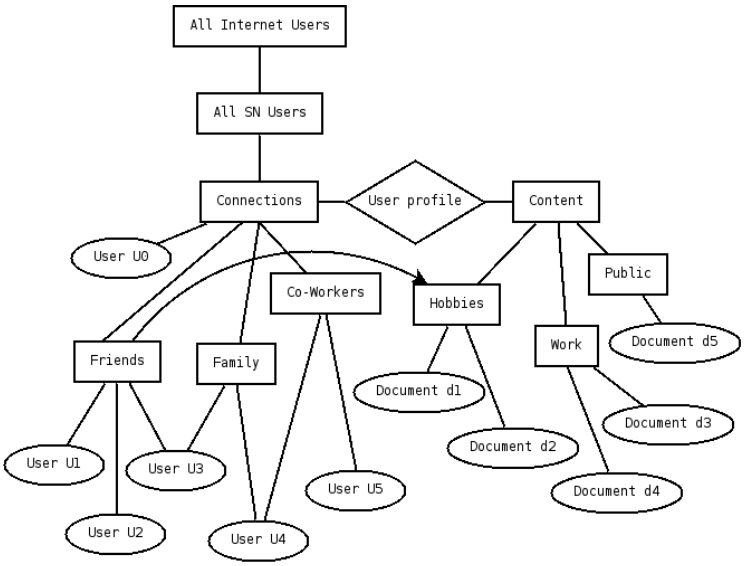

Fig. 1. Our approach with one access control example scenario

Figure 1 represents an example of our approach, where the user's profile is divided into Connections and Content Classes, that are further divided into sub-groups of classes, forming a hierarchical structure. The graph with the social network users' connections structure is represented in the social network provider servers. However, the social network user's trusted circle of friends, and the

respective public keys are controlled on the client side by the user himself. The classification of friends into groups can thus differ from that stored on the social network site himself.

With the approach presented in Figure 1 it is possible for the social network user to define his privacy statements and restrict access to documents or content classes to a set of connections (users or classes).

When a social network user posts new content he makes a selection within his connections classes, that are represented in his local trusted circle structure, to specify who will have access to the content. In this way, the user keeps his personal data private for a pre-defined audience.

An example of such audience segregation is presented in Figure 1. All members of the 'Friends' class have access to all the documents from the social network user's 'Hobbies' class.

This is possible because all the members that belong to the group of Friends of the social network user share the secret to retrieve the content. Also note that users and data can reside in several connection/content classes simultaneously.

4.3 Access Control Enforcement

We propose in our model to use cryptographic techniques in order to enforce access control. In our prototype application we use OpenPGP[7] standard to keep social network users' data confidential. One nice feature of OpenPGP is its support for encrypting to multiple recipients using hybrid encryption, by encrypting the content with a random secret and the secret with all the public keys of the set of users. We assume that each user holds a public and secret OpenPGP key pair. Whenever a new connection between two social network users is established, these users exchange their public keys. The shared public keys are then stored locally and compose the user's circle of trust. The OpenPGP public key can also be retrieved from an online key server by name or email mapping.

As an example of the flow, let Alice and Bob be two users in a social network site. Bob accepts Alice as his friend. He then adds Alice's public key to his key-ring, and including Alice in a circle of trust. Then, Bob can post encrypted messages that can only be accessed by a selective audience chosen from the Bob's circle of trust.

In further stages, we plan to use more advanced cryptography, similar to approaches presented for other contexts in [5] [10] that use hierarchical encryption. We also plan for better integration of the tools that are provided in the social network sites together with our model for better configuration options.

4.4 Implementation

In order to provide users with an access control tool we have built a prototype implementation. The enforcement of the access control is done by using the OpenPGP encryption protocol, whereas the definition of the access control is

[7] http://www.openpgp.org/

done by using an user-interface, which needs some improvements before being available to the open-source community. Additionally, improvements may be required on the access control enforcement by using more advanced cryptographic techniques.

Technical Details To simulate a real world social network, a social network testbed site was created using the Elgg [8] open source framework. Elgg is an open source social networking platform with a large user base and is written in PHP.

The prototype application was developed as a Firefox extension, that allows client-side access control enforcement for independent platforms. The approach of using a firefox extension is related to the extensive usage of the browser and for the flexible integration mechanisms available.

The Firefox extension development is done by using XUL, CSS and JavaScript, and can also take advantage of XPCOM[9] technologies for more sophisticated features, which is a cross platform from Mozilla that allows integration between different technologies. The encryption was done using a GPG library from FireGPG [10] in Javascript that accesses a IPC binary library.

Prototype Architecture With the prototype extension, users are able to execute control over their data with no third-party influence. This is managed by each user by having a local trusted circle. Thus, the selected members of that circle can be allowed to have access to a target content, like an access control list.

The high-level architecture of our application is represented in Figure 2, whereas a simple flow of the application is shown in Figure 3. Each user has to create their circle of trust. In our initial approach we consider that users exchange the OpenPGP public key when a SNS friendship connection between them is created. The group and key management is afterwards done locally by the application and managed by the user. Therefore, when the user adds new data into the SNS some privacy options are given by the application in order to automatically define the selective list for the content to be posted. The access control can then be enforced for a selective individual(s) or group(s) in the trusted circle and posted into the SNS. For reading protected content that has been posted in the SNS, the user has to be given access by the content owner when the content has been posted. The prototype application parses the website and searches for encrypted, OpenPGP, blobs of text. Then, if the user has read access the application automatically decrypts the content and presents the unencrypted data to the user, by rewriting the webpage. Otherwise, the absence of clear-text data is indicated by a pre-defined message, like *Non-authorised content*.

In terms of access control enforcement, the Firefox extension features an user-transparent application, using esatblished cryptographic techniques (OpenPGP)

[8] http://elgg.org/
[9] http://www.mozilla.org/projects/xpcom/
[10] http://getfiregpg.org/

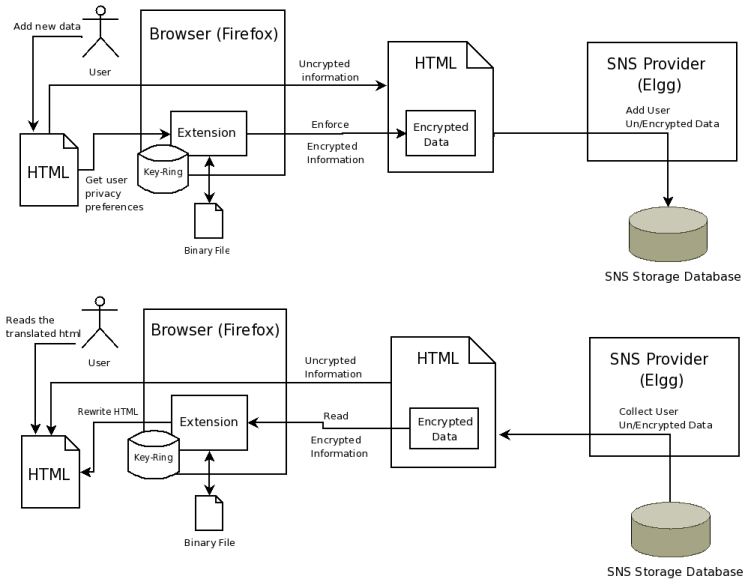

Fig. 2. High-level architecture (Top: Post process; Bottom: Get process)

to enforce the access control defined on the page content. It allows encryption and decryption of one-to-one and one-to-many relations (like groups). This is an useful functionality for multiple recipients encryption, such as groups or multiple users. However, the length of the output will be also directly affected by the increase of users.

5 Conclusion

We designed and implemented a system that allows users to define and enforce selective access control policies for data published on social network sites. By using a PKI encryption scheme, such as OpenPGP we were able to keep users' data confidential, even towards the SNS operator, by means of encryption. Through the integration into a Firefox extension encrypted content is automatically decrypted by the browser of authorised users. The extension also allows for the definition of groups and for the encryption of content under the keys of all group members. Our extension is simple and aims at striking the difficult balance between usability and privacy for general users. We tested our extension with our own social network site tesbed and in other social network sites, like Facebook and MySpace.

Due to the fact that it has been design to be general and SNS independent, it is also possible to use in other Web 2.0 solutions, such as blogs, forums and wikis. Therefore, an improvement on the user interface and on the efficiency of the prototype application is important.

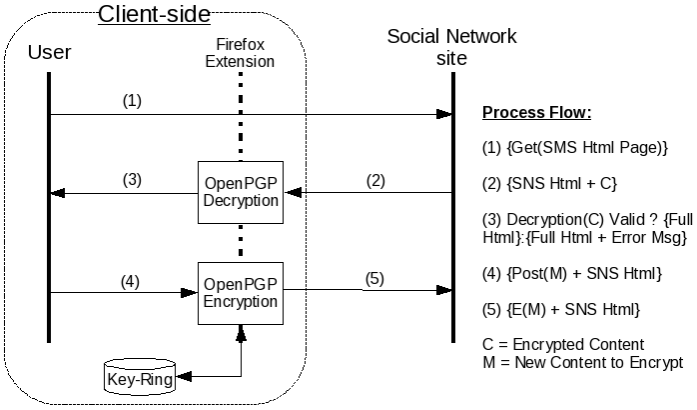

Fig. 3. Prototype application Flow

6 Future work

As future work we plan to conduct user studies to further improve our interface. Moreover, we plan to extend our prototype to support more advanced policy concepts and cryptographic techniques for access control enforcement.

One weakness of the client side implementation is that the user needs to duplicate the friendship relations that he establishes in different social networks locally, on his machine. In the future, we plan to support synchronisation mechanisms with different social network sites that synchronises the client side friendship definitions with those of various social network sites. This should also include support for key management to obtain the necessary OpenPGP keys. In this way, once the user adds new friends to one of his social networks, the Firefox extension should update it's state accordingly.

Moreover, the current plug-in is oblivious of relations between different content items. Thus the user himself is responsible for encrypting all content items related to 'Hobbies' to his 'Friends' connection class. In the future, the extension might be able to derive this information from social network specific tags that are added to the content or that can be derived from the context.

Cryptographic issues of the current system are the linear growth of the ciphertext with respect to the size of the connection classes, and the fact that OpenPGP encryption is not anonymous.

The first issue is related to the use of hybrid encryption. Hybrid encryption adds one public key encryption of the symmetric key for each recipient. One approach for reducing the ciphertext size is the use of a broadcast encryption scheme [4]. Another approach is to reuse symmetric keys for multiple documents with the same access rights, or to use hierarchical key derivation [5].

The second issue is related to the way in which the encrypted blocks are decrypted. An OpenPGP ciphertext contains a 64-bit Key ID of the recipients

public key. Thus while looking rather random, everyone can determine the recipients of the ciphertext. While this speeds up the test to determine whether or not the extension can decrypt a OpenPGP block, this reveals privacy sensitive information. Anonymous encryption [6] does not reveal which public key was used to create the ciphertext. The only way for determining whether a user is among the recipients would be to trial decrypt on average half of the public key encrypted blocks of the hybrid encryption. This is inefficient but more privacy friendly.

References

1. *(Under)mining Privacy in Social Networks*, 2008. Google Inc.
2. Alessandro Acquisti and Ralph Gross. *Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook*. 2006.
3. Geoff Salmon Amin Tootoonchian and Ahmad Ziad Hatahet. Fine grained access control in online social networks. Technical report, 2007.
4. Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In *ASIACRYPT '01: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security*, pages 566–582, London, UK, 2001. Springer-Verlag.
5. C. Blundo, S. Cimato, S. De Capitani di Vimercati, A. De Santis, S. Foresti, S. Paraboschi, and P. Samarati. Efficient key management for enforcing access control in outsourced scenarios. In *Proc. of the 24th IFIP TC-11 International Information Security Conference (SEC 2009)*, Cyprus, Greece, May 2009.
6. D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Crypto*, pages 258–275, 2005.
7. Erving Goffman. *The Presentation of Self in Everyday Life*. Doubleday, Garden City, New York, 1959.
8. R. Gross and A. Acquisti. Information revelation and privacy in online social networks (the Facebook case). In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80.
9. Saikat Guha, Kevin Tang, and Paul Francis. Noyb: privacy in online social networks. In *WOSP '08: Proceedings of the first workshop on Online social networks*, pages 49–54, New York, NY, USA, 2008. ACM.
10. Himanshu Khurana, Adam Slagell, and Rafael Bonilla. Sels: a secure e-mail list service. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 306–313, New York, NY, USA, 2005. ACM.
11. Balachander Krishnamurthy and Craig E. Wills. Characterizing privacy in online social networks. In *WOSP '08: Proceedings of the first workshop on Online social networks*, pages 37–42, New York, NY, USA, 2008. ACM.
12. Matthew M. Lucas and Nikita Borisov. Flybynight: mitigating the privacy risks of social networking. In *Proceedings of the 7th ACM workshop on Privacy in the electronic society (WPES)*, pages 1–8, New York, NY, USA, 2008. ACM.
13. Marco Casassa Mont, Siani Pearson, and Pete Bramhall. Towards accountable management of identity and privacy: Sticky policies and enforceable tracing services. pages 377–382. IEEE Computer Society, 2003.
14. Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. Technical Report arXiv:0903.3276, Mar 2009. Comments: Published in the 30th IEEE Symposium on Security and Privacy, 2009.

15. Frederic Stutzman. An evaluation of identity-sharing behavior in social network communities. In *Journal of the International Digital Media and Arts Association*, 2006.