# Privacify: Extending Privacy in Online Social Networking

Rodrigo Pereira Botelho, Sérgio Donizetti Zorzo

Distributed Systems and Networks Departament
Federal University of Sao Carlos
Sao Carlos, Brazil
rodrigo_botelho@dc.ufscar.br
zorzo@dc.ufscar.br

**Abstract:** Online social networks typically provide tools for users to set who can access their shared data. However, this access restriction only applies to network users and not for third parties and the social network itself. An approach only with data encryption is insufficient to keep both data privacy and the user's ability to obtain personalized services. This paper presents a model to extend the privacy in online social networks, ensuring the privacy of certain data from other network users, third parties and own social network, yet allowing the use of personalized services for users of social networking.

## 1 Introduction

Users of online social networks tend to accept the privacy policies of these sites so they can share/exchange information with people they know who also use such a service (friends who use the social network). All information shared is held by the entity that operates the social network and can be used for various purposes, for example, providing personalized service improvements, advertising, among others.

As the success of most social networks depends on the users' satisfaction regarding the use of the service, the social network operator ensures to protect the privacy of the shared data so that data are not accessed and misused by third parties. To this end, the social networking sites provide some tools for users to cancel or grant access to specific data in their profile, for example, a tool for determining the visibility (public/private) photo album, message board.

However, few or no social network provides similar artifacts to control third party access to such data and without this control the use of aggregate data from users for advertising purposes is common. Although the user probably has agreed to a privacy policy that prescribes this kind of use, it is not known for sure if only aggregate data, and not sensitive information, is actually being shared with these third parties. Another important point is that installed social applications (games, utilities) can access certain data from the user profile such as politcal views, sexual orientation and list of friends.

It can be noticed that the tools that enable controlled access to data provided by social networks are elementary and only control what other network users can view and not the access and use of this data by third parties. From the user perspective, it is interesting that the access control to data is done by the user. This means that data are not shared with the network or with third parties if the user does not want.

The previously mentioned scenario is not realistic, given that the format we have today is ideal for social networking, because taking possession of the user data social networks can aggregate this information and use it in various ways. Thus, a balance to satisfy both, users and social networks, leads to a model in which users can protect their data by making them visible only to those who is their interest and that social networks can benefit from aggregate data from users.

In this paper we present a model to ensure that user can keep some personal data private while allowing this user to receive personalized services, as well as a proof of concept implementation. The rest of the paper is organized as follows: section 2 presents related works, in section 3 the model/architecture is presented, section 4 presents the implementation, section 5 presents an analysis of the model and the implementation and in section 6 the conclusion is presented.

## 2 Related Work

In recent years many studies have been developed to improve privacy in social networks. Problems of re-identification from social networks structure [5-7], shared data between social networks and social applications [8], security breaches [9, 10] and privacy policies [4, 11-13] are among the topics discussed in the researches.

Besides the previously mentioned studies, other studies are focused on how the data that users post on social networks are processed and shared. It is not intended to cite a complete list of papers, but compare this work with some previously proposed approaches.

In [4], Kodeswaran and Viegas propose a policy-based infrastructure to provide access to social network data, keeping users' privacy. The authors created different types of access control and if compared with the traditional approach allow/deny access to certain data allows more expressiveness when you specify a type of access. This makes possible, for example, certain kinds of user data to be aggregated for different purposes, even if the user's sensitive data are not disclosed it is possible to ensure the user privacy while being able to provide personalized services. However, the proposed architecture requires a trusted server that stores user data and provides the right type of access based on policies, and if this server is compromised, the privacy of users shall be as well.

The Lockr [2] uses social attestations and access control list to determine whether a user can or cannot access a specific content. This allows a user's social network to be maintained offline, without the need to use any social networking service.

The proposal of Baden *et al.* [3] allows through attribute-based encryption (ABE) that the users and not the social network to control who accesses data. The proposed architecture requires some additional components to perform the operations of writing and reading of data and maintain the friendship relation of social network users.

FlyByNight [1] describes a system that uses encryption to keep user data private. The data are encrypted before being sent to the social network, ensuring privacy even if they are publicly visible on the social network. The paper describes a different approach to that presented in Persona [3] for sending messages to groups using a proxy-based encryption approach rather than on attribute-based encryption.

FlyByNight [1], Lockr [2] and Persona [3] use encryption to protect data, differing mainly in how each one gives a certain group of users access to data. Thus, even if the server that stores the data is compromised the privacy of user data is guaranteed. However, neither approach provides a method similar to Kodeswaran and Viegas [4] to aggregate user data and to provide customized services. A combination of features leads to the model presented in section 3.

## 3 Model

Today social networks receive a large amount of data generated by users. These data are photos, personal data, interactions between users, among others. The risk involved in sharing these data is the maintenance of them because once the data were shared on the social network can occur a situation in which the user can not have control over the use of these data. For example, a copy of the data can be made for third party services.

The model proposed by this paper makes use of encryption methods to encrypt data before it is sent to the social network. Thus, even if the data is copied to third-party services will not be very usefulness, since a recipient must know the keys to be able to read a message content.

Some requirements are essential to make the model utilization in practice to interfere as little as possible on how a user uses the network. One of these requirements is the user ability to use this proposed solution from any computer. Another important requirement is to not trust the social network server even if it can be trusted. Thus, the data is shared only by those who were initially assigned by the data owner. Finally, a user should be free to return the normal use, without adding protection to data.

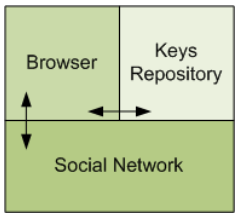


Figure 1: Model components

Figure 1 shows the components that make up the proposed model. The *Social Network* component is the representation of any social network which the *Privacify* will be applied. The *Browser* component is the representation of tool or program which users access and use the social network. This component is responsible for preparing the messages in the **`Privacify-Message`** format before it is sent to the social network. Finally, the *Keys Repository* component is a representation of a service that maintains information of user keys and the keys of the user's friends. These keys are used to encrypt the message and then to read the encrypted message.

The components *Browser* and the *Keys Repository* are trusted elements in the model and the arrows in Figure 1 indicate the directions in which communication can occur. Note that the *Social Network* component does not communicate directly with the *Keys Repository*.

In a simplified manner, to a user be able to communicate using the *Privacify*, the user must generate a pair of public/private key and obtain the public keys of all users to whom he wants to maintain communication. These keys are stored in the *Keys Repository* and every time the user is using the *Social Network* these data are loaded into the *Browser*. Is the role of the *Browser* component, ensure that those keys are provided with security and privacy for the *Keys Repository*.

Every message sent to the Social Network server is on **`Privacify-Message`** format, which is illustrated in Figure 2.

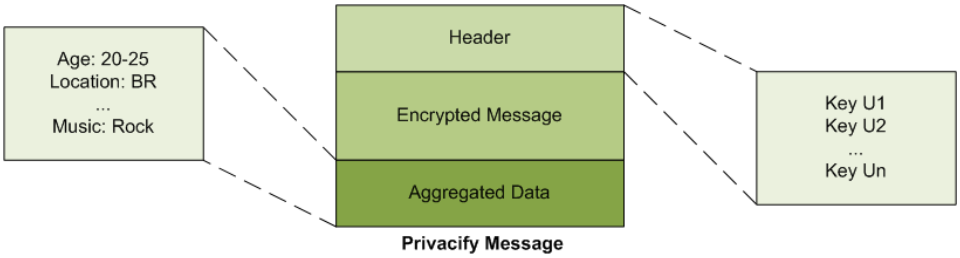


Figure 2: Message format

The message can be divided into three sections: *Header*, *Encrypted Message* and *Aggregated Data*. The *Header* section contains information for that authorized users can read the message. In other words, it must contain sufficient information to each related user can be able to decrypt the cipher text. The *Encrypted Message* contains the payload of the message. To support advertising and access to specific data through social applications, the *Aggregated Data* field was added. With this field it is possible to aggregate some sensitive information, so the exact values are not revealed. For example, instead of providing precisely the age, we can put an age range in the *Aggregated Data* field.

The model supports both messages sent to a single user or for multiple users. The difference between the two types of messages is the number of users listed in the message *Header*. The cipher text is unique not being necessary to encrypt the message *N* times to send to *N* users, which would make the model implementation prohibitive for reasons of overhead in message size. The message overhead is discussed in Section 5.

It is important to note that the proposed model does not guarantee the total privacy of user data. Social connection data, such as friends list, are still visible to the social network. However, ensures additional privacy through encryption of data that are posted explicitly. This additional privacy protects data from social networking and other sources, if any leaks.

Figure 3 shows the levels of privacy that can be obtained with *Privacify*.

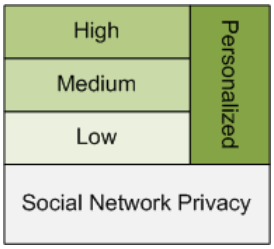


Figure 3: Privacy Levels

At the lowest layer is the level of privacy provided by the actual social network. As previously mentioned, this level may include access control in parts of user data. But this control is in relation to other users and not for third parties. The layers *Low*, *Medium*, *High* and *Custom* relate to levels of privacy provided by *Privacify*. With the exception of *Custom*, the top layers always provide protection from lower layers.

The *Low* layer of privacy only protects the user profile data, *e.g.* name, email, age, political views, among others. Some of these data, however, can be provided in the form of aggregated data for the social network in this way the services already offered are not harmed. *Medium* layer protects all text-based messages, for example, comments on photos, testimonials, among others. The protection provided by the *High* layer goes beyond text messages, providing privacy for all data posted by the user, for example, photos, videos and more. Finally, the layer *Custom* user can choose which data you want to keep private.

By observing the levels of privacy in Figure 3, one can see that the *Privacify* can be used to extend the privacy of online social networks supporting data privacy as well for third parties including the social network itself. To illustrate, suppose that a social network that allows users to change the policy on access to their profile data to “public”. If user set up privacy level as *Low* on *Privacify* his profile data may also be retrieved by all users of the network, but only authorized users will be able to read the content so we are extending the privacy of social network.

## 4 Implementation

The model proposed in section 3 was implemented using Orkut as the *Social Network*, chosen for its popularity in Brazil [14]. For the other two components of the model were developed an extension to the Web browser Google Chrome and a key repository that accepts and responds RESTful requests keys implemented in PHP.

To use the browser extension (*Privacify-SN*) for the first time it is necessary to generate the pair of public and private user, which will later be stored in the key repository *Privacify-Service*. To avoid some types of attacks intended to steal the private keys of users during transmission to the repository, when generating the key pair each user enters a password that is used to encrypt their private key with a symmetric key encryption before the keys are transmitted to the repository. Moreover, the password is not transmitted to the repository and in some cases is not even stored on the user's computer and he should just remember the secret used initially in successive accesses to that his private key can be decrypted and used.

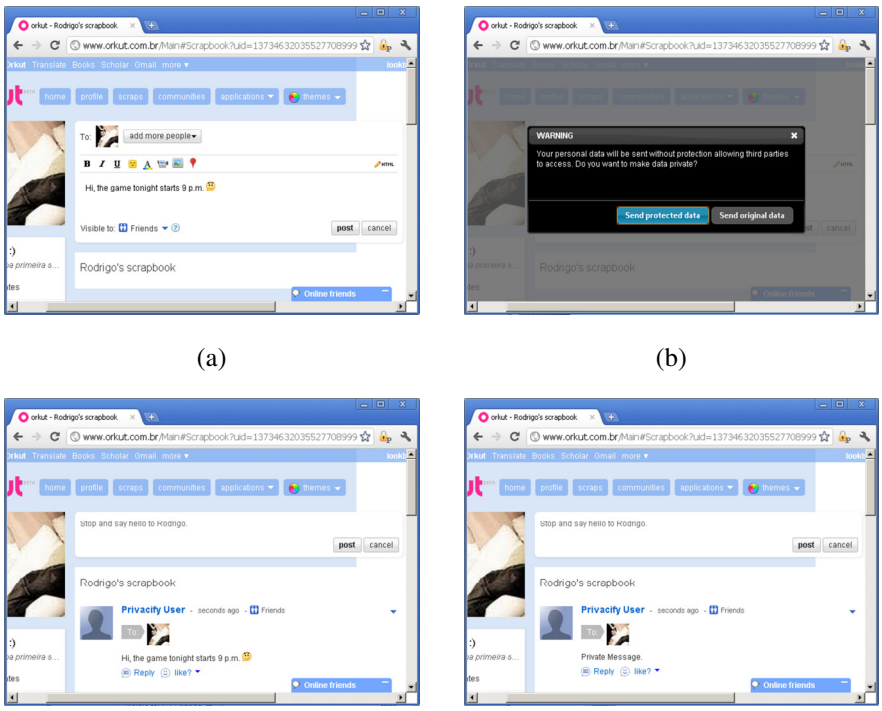


Figure 4: Sending and viewing a message with Privacify-SN

After the generation of keys the user is almost ready to send a message using the *Privacify-SN*. The next step is to get the friends public keys for some disconnected means. This is a laborious process but necessary to ensure that safety. With the possession of keys, the user can then send messages using the *Privacify-SN*.

Figure 4 shows the steps to send a message and how this message is displayed. In this figure the user is using a default configuration, then after typing a message to his friend, Figure 4 (a), he clicks the post button. Then a *Privacify-SN* popup window opens stating that the data will be sent without protection and asking if the user wants to add data protection. If the user chooses to add protection, public key from source user and destination user is gathered. After this, a random secret is generated and used to encrypt the original message by means of symmetric key encryption, which is added in the body of *Privacify-Message*. Then, the random secret is encrypted with the public keys of the users involved through asymmetric key cryptography and added appropriately to the message header. As in this case there is no aggregate data, the *Aggregated Data* field of *Privacify-Message* is blank. So instead of sending the original message to the social network the *Privacify-Message* is sent.

Reading a private message is done similarly. When accessing a page with protected content, the *Privacify-SN* reads the message header identifying what messages are intended for the current user which is using the *Privacify-SN*. If it finds any message intended for the current user, Figure 4 (c), the user's private key is used to decrypt the secret which was used to encrypt the message and then the original message is displayed to the user. Otherwise, a default message "Private Message" is displayed, Figure 4 (d).

## 5 Analysis

This section examines some points of the proposed such as the binary data can be treated, size overhead in messages and information which remain visible for the social network.

Some social networks allow users to share data which are not necessarily plain text. For example, it is common in social networks to find a photo album feature for photo sharing. In this case, binary data can be protected by converting them to a textual representation, for example, base64. After this step we can encrypt the text representation as is done with the other texts, adding the cipher text properly on *Privacify-Message*.

Each message sent by *Privacify-SN* consists of a header, a cipher text and optionally by a aggregate data field. Thus, one can see that each message has an overhead because besides the original message other information is sent by the *Privacify-SN*. The header overhead is directly related to the size of RSA keys that are used to encrypt the secret message. As we use a fixed size for the random secret used to encrypt the original message, the header overhead is presented in accordance with Table 1.

**Table 1.** Message header overhead.

| RSA key size (bits) | Overhead (bytes) |
|---|---|
| **512** | 128 * N |
| **1024** | 256 * N |
| **2048** | 512 * N |

The amount of overhead in bytes is multiplied by *N* because *N* represents the number of recipients which the message was sent. Table 2 shows the overhead of the encrypted message regarding the original message. The original messages are texts dealing with social networks typical greetings like "Hi, how are you? What are you doing tonight?" Because it is text-based messages, the overhead imposed can be easily mitigated if the storage server utilizes some text compression technique before the storage is made.

**Table 2.** Overhead in the cipher text.

| Original Message (bytes) | Encrypted Message (bytes) |
|---|---|
| **60** | 145 |
| **161** | 281 |
| **285** | 450 |

Finally, one important aspect about the purpose of this study is that although the approach enables to increase the level of privacy while maintaining the ability to offer personalized services through encryption and a field of aggregated data, some information is still visible to a social network, as is the case of friendships - social graph.

## 6 Conclusion

This paper presented a model and implementation for extending privacy in online social networks. It was observed that the proposal improves privacy while maintaining user's ability to receive personalized services, but has the limitation of not providing total privacy leaving some information still visible to the social network operator, as is the case of friendly relations - social graph.

The design combines encryption with a well known message format. This way, a user can selectively choose which users can see some personal data. Due the aggregate data field in the *Privacify-Message*, online social network operators and third parties can keep offering personalized services while the access to the real data is kept private through cryptography. This way, it can be seen that user data maintained private but the interactions between users are not. As it is believed that there is no perfect privacy with personal data revelation, granting access just to a piece of aggregate data seems to be a valuable alternative.

Moreover, the proposal keeps the data private even though the storage server is compromised. To this end, an overhead is added to each private message sent. However, as discussed, this is not a prohibitive limitation and therefore the proposed approach is shown possible to be implemented in real environments.

## References


1. Lucas, M.M. and N. Borisov, *FlyByNight: mitigating the privacy risks of social networking*, in *Proceedings of the 7th ACM workshop on Privacy in the electronic society*. 2008, ACM: Alexandria, Virginia, USA.
2. Tootoonchian, A., et al., *Lockr: social access control for web 2.0*, in *Proceedings of the first workshop on Online social networks*. 2008, ACM: Seattle, WA, USA.
3. Baden, R., et al., *Persona: an online social network with user-defined privacy*, in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*. 2009, ACM: Barcelona, Spain.
4. Kodeswaran, P. and E. Viegas, *A Policy Based Infrastructure for Social Data Access with Privacy Guarantees*, in *Proceedings of the 2010 IEEE International Symposium on Policies for Distributed Systems and Networks*. 2010, IEEE Computer Society.
5. Zhou, B., J. Pei, and W. Luk, *A brief survey on anonymization techniques for privacy preserving publishing of social network data.* SIGKDD Explor. Newsl., 2008. **10**(2): p. 12-22.
6. Hay, M., et al., *Anonymizing Social Networks.* SCIENCE, 2007. **245**: p. 17.
7. Ahn, Y.-Y., et al., *Analysis of topological characteristics of huge online social networking services*, in *Proceedings of the 16th international conference on World Wide Web*. 2007, ACM: Banff, Alberta, Canada.
8. Felt, A. and D. Evans, *Privacy Protection for Social Networking Platforms*, in *WEB 2.0 SECURITY AND PRIVACY 2008*. 2008: Oakland, CA, USA.
9. Kaafar, M.A. and P. Manils, *Why spammers should thank Google?*, in *Proceedings of the 3rd Workshop on Social Network Systems*. 2010, ACM: Paris, France.
10. Huber, M., et al., *Exploiting social networking sites for spam*, in *Proceedings of the 17th ACM conference on Computer and communications security*. 2010, ACM: Chicago, Illinois, USA.
11. Wu, L., et al., *Analysis of social networking privacy policies*, in *Proceedings of the 2010 EDBT/ICDT Workshops*. 2010, ACM: Lausanne, Switzerland.
12. Toch, E., N.M. Sadeh, and J. Hong, *Generating default privacy policies for online social networks*, in *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*. 2010, ACM: Atlanta, Georgia, USA.
13. Squicciarini, A.C., M. Shehab, and J. Wede, *Privacy policies for shared content in social network sites.* The VLDB Journal, 2010. **19**(6): p. 777-796.

14. comScore. *Orkut Continues to Lead Brazil's Social Networking Market, Facebook Audience Grows Fivefold*. 2010 [cited 2011 March]; Available from: http://www.comscore.com/Press_Events/Press_Releases/2010/10/Orkut_Continues_to_Lead_Brazil_s_Social_Networking_Market_Facebook_Audience_Grows_Fivefold/(language)/eng-US.