Preserving the User's Privacy in Social Networking Sites

Alexandre Viejo, Jordi Castellà-Roca, Guillem Rufián

Departament d'Enginyeria Informàtica i Matemàtiques,
UNESCO Chair in Data Privacy, Universitat Rovira i Virgili
Av. Països Catalans 26, E-43007 Tarragona, Spain
Corresponding author: alexandre.viejo@urv.cat

Abstract. In the last years, social networking sites (SNSs) have enjoyed an undeniable success. Those web platforms have huge quantities of active users sharing lots of information everyday. Usually, user-generated content may be almost innocuous, however, some studies have shown that it may also contain very sensitive personal data. This situation may pose a serious privacy threat to the users due to the fact that third parties can gather and exploit that knowledge for their own benefit. There are some proposals in the literature that try to address this situation. Nevertheless, they fail to provide a practical solution capable of working with well-known SNSs. In this paper, we propose a new scheme that fills this gap. More specifically, we present a privacy-preserving system that enables the users to decide which individuals (e.g., other users, third parties or even the SNS itself) can access to their user profiles. We have implemented our scheme to be used by Facebook users. We have run some tests with our prototype and the results show that the added overhead is affordable.

Keywords: Privacy, Confidentiality, Social Networks, Access Control, Facebook.

1 Introduction

Social networking sites (SNSs) are the most representative result of the rise of the Web 2.0 and its related technologies. In these environments, users publish and share information and services that can be easily accessed by a global audience.

The success of these platforms can be effectively measured in terms of number of users, and the results are really stunning. Specifically, main players like Facebook or Twitter claim to have more than 800 and 100 million active users respectively [1]. More impressive is the fact that those numbers grow each day and their limit cannot be still envisaged.

With such a huge quantity of users and so many different activities available on SNSs, the amount of user data which can be gathered from those places is especially large and heterogeneous. Particularly, user-generated content may reflect general opinions and information which can be considered innocuous but it also might contain very sensitive personal data. In this way, the *Consumer Reports'2010 State of the Net analysis* [2] states that more than half of users of social networks share private information about themselves online.

The existence of sensitive information among the data publicly shared by the users may represent a relevant privacy threat due to the fact that third parties can gather and exploit that knowledge for their own benefit. More specifically, leakage of personal data, especially one's identity, may invite malicious attacks from the cyberspace (*e.g.*; personalized spamming, phishing, etc) and even from the real world (*e.g.,* stalking) [3].

Recently, these privacy concerns have been reported to negatively affect the way the users use SNSs. In this way, a survey presented in [4] shows a strong association between low engagement and privacy concern. Specifically, users who report concerns around sharing control, comprehension of sharing practices or general SNS privacy concern, also report consistently less time spent as well as less posting, commenting and "Like"ing of content. This situation can be harmful for the SNSs since their business model requires large quantities of users generating new content without limit.

Therefore, in the last years, the SNSs themselves have provided some privacy settings for their users that allow them to set the privacy level of their online profiles and to disclose either some or none of the attributes in their profiles [5]. However, this privacy-preserving approach suffers from two main problems: (i) these privacy settings are generally not sufficiently understood by the average users who seldom change the default configuration [6](according to [7], this configuration generally makes most of the user information public [7]; and (ii) this method does not prevent the SNS itself from gathering the sensitive user data, in fact, a relevant percentage of the users are worried about how SNSs protect their privacy [8] due to the fact that they are aware of their data being exploited by advertisers [9].

Due to the fact that the companies that support SNSs are not fully reliable in terms of protecting the user's privacy, in order to limit the privacy problems that have been stressed above, it is necessary to design new privacy-preserving mechanisms intended to be deployed and managed by the users themselves.

1.1 Contribution and plan of this paper

In this paper we propose a new scheme that enables the users of SNSs to decide exactly which individuals can access to their published sensitive data. This implies that other users, third parties or even the SNS itself cannot obtain any protected information if this is not explicitly allowed by the owner. Obviously, this approach does not rely on the active collaboration of the SNS.

The target platform of this proposal is a typical SNS where the user has a profile, a list of friends and a place to publish photographs or images (e.g., Photo Album or similar). Even though the proposed mechanism can be deployed in any SNS that fulfills those requirements, in this work, we have implemented it to be used with Facebook.

Section 2 introduces the state of the art related to the privacy-preserving approaches which can be found in this field of research. Section 3 introduces the system model. Section 4 details our new proposal. Section 5 evaluates the runtime cost of the proposed scheme. Finally, Section 6 reports some concluding remarks.

2 Previous work

The use of user privacy policies (in the form of a contract) to ensure the proper protection of private data is one of the main approaches to provide privacy-preserving SNSs. For example, the works presented in [10–12] follow this idea. The main shortcoming of these proposals is that SNSs are supposed to implement such policies to improve the privacy of their users and this is currently unrealistic. Under this research line, it is worth to mention the existence of Persona [13], a social network integrated with Facebook as an application to which users log in through a Firefox extension. In Persona, users define a privacy policy that manages access to their information. As a result, only users of Persona with the necessary access rights can get the protected data. Nevertheless, this tool is only a Facebook application that can be easily removed by Facebook from the applications directory.

Other researchers have focused on designing new SNSs that effectively address the privacy concerns of the users. These platforms generally trust on a completely distributed architecture. Diaspora [14] is a clear example of that. This SNS is a privacy aware, personally controlled, do-it-all distributed open source social network. This project is described as a network that allows everyone to install their own "seed" (*i.e.*, a personal web server used to store photos, videos and everything else) within the larger network. That seed is fully owned and controlled by the user, so the user can share anything and still maintain ownership over it. In this way, the social network gives individuals control over their personal information without being subjected to changing privacy policies and sell-outs to third parties [15]. Diaspora is not the unique system that follows this approach. Other privacy-preserving SNSs based on p2p architectures have been proposed in [16–18]. However, the main drawback of all these systems is that they will be hardly adopted by the mainstream audience. Note that centralized SNSs like Facebook and Twitter are very well established and it is quite unrealistic to assume that a new competitor without a very strong company behind will get enough users to represent a proper alternative.

Focusing on privacy-preserving approaches that can be integrated with traditional SNSs, a straightforward solution to prevent any unauthorized entity from accessing the protected user data is using cryptography primitives to cipher any text or attribute before publishing it. Applying this method, only the individuals with the correct cryptographic keys will be able to access the protected content. Nevertheless, this solution is quite problematic because, usually, registering on well-know social networks under a pseudonym, or obfuscating personal information in any way is forbidden by the terms of service. More specifically, Facebook (the target platform of this work) has banned users who have violated those terms [19]. According to that, the ideal privacy-preserving method should generate protected data that does not look suspicious to the SNS, this is, the information to be published must look real while being incorrect (or, at least, partially incorrect).

Following this idea, the authors in [20] present a scheme that, first, divides the private data into atoms and, then, replaces each atom with a corresponding atom from another randomly selected user who uses the same application. Two significant shortcomings of this proposal are: (i) it requires a certain number of users to provide anonymity; and (ii) it requires some external infrastructure that keeps the relations between the users and their atoms.

A similar proposal is introduced in [21]. This is a Firefox extension that allows users to specify which data or activity need to be kept private. The sensitive data is substituted with fake one, while the real data is stored in a third party server that can be only accessed by the allowed users. Like in the former proposal, one of the main shortcomings of this scheme is that it relies on a centralized infrastructure that must be honest and always available.

Finally, [22] addresses the problem of the centralized infrastructure by locally storing the real data on the allowed friends' machines. In this way, only fake information is stored on Facebook. When a user using this scheme browses a profile of another user who also uses this system, a software component is in charge of transparently showing the real information stored locally, instead of the one actually published on the SNS. This solution requires the users to always connect to the SNS using the computer that locally stores the real data. This may be a main problem for certain users. Moreover, whenever a certain user modifies her protected information, it has to be individually sent to all authorized friends. This issue is not quite efficient in terms of bandwidth usage and it might generate some unstable situation where not all the authorized recipients would have access to the newest information. Also, this solution requires the users to store in their own computers unspecified quantities of information related to others. Some users may feel uncomfortable with this situation, while others might not be willing to spend their storing resources on this task.

In order to solve all these issues, the authors in [22] propose to store all the protected information steganographed within images published in the SNS. Even though this idea is quite promising, the authors do not develop it in their work and it is even not considered for future work.

3 System model

As explained previously, we propose a new privacy-preserving scheme that enables the users of SNSs to decide exactly which individuals can access to their published sensitive data.

We next detail the kind of SNSs which can be the target of our proposal. Then, the requirements of the designed system are provided. Finally, we briefly describe how our system works and its architecture.

3.1 Target SNS

Our work has been designed at high level to be integrated with any SNS that offers the following assets: (i) a user profile; (ii) list of friends; and (iii) place to publish photographs or images (e.g., Photo Album).

Due to the fact that Facebook is a really well-known SNS that properly fulfills all those requirements, we have chosen this platform to implement our proposal and retrieve some empirical results. Accordingly, Facebook is considered the target SNS in the rest of this document.

3.2 System requirements

At the current development stage, the main target of the proposed scheme is to enable users to only protect their personal data which appears in their "User Profile" section of the SNS. This implies that hiding other sources of information such as the list of friends or the timeline/wall (i.e., a section of the SNS where users and friends publish text and images) is left for future work.

A complete user profile in a SNS such as Facebook reveals a lot of sensitive information from the owner: gender, date of birth, current location, religious or political views, current and past jobs, interests, education, marital status, etc. This fact clearly stresses the relevance of preventing any unauthorized entity from freely gathering information from this source of data.

Personal data must be protected but this must be done in a transparent way from both the point of view of the users and the SNS. In the case of the users, nowadays, a huge quantity of them are already used to interact with classic SNS (like Facebook) in a determinate way. Therefore, in order to be fully adopted, any privacy-preserving solution should not interfere (or interfere the least possible) in the fixed routine of the users. Regarding the point of view of the SNS, we have explained previously that Facebook (or other similar platforms) does not allow its users to publish fake information in their accounts. Therefore, in order to reach its target, the privacy-preserving mechanism must publish fake data that looks real in front of the SNS.

3.3 Our scheme in a nutshell

The main idea behind the proposed system is to replace the sensitive data that can be found in the "User Profile" section of a SNS with fake information introduced by the user herself. The proposed scheme first uses cryptography to protect the original sensitive information and, then, it hides the ciphered data in a certain image by means of steganography. Access-control techniques are applied to allow only certain users to retrieve the original information. The resulting image is finally published in the place reserved by the SNS to publish images (e.g., Photo Album).

When any entity (e.g., users, external third-party, the SNS itself) tries to read the "User Profile" of a protected user, two main situations may apply depending on whether this entity is aware of the privacy-preserving system used or not:

- *The reader is not aware.* In this case, this entity only obtains the fake information introduced by the user who runs the privacy-preserving method. If the target SNS does not allow users to obfuscate their personal information, the introduced fake data must look real in order to fool it.
- *The reader is aware.* In this case, the reader looks in the Photo Album for the image that contains the real information (i.e., the stego-object), obtains the ciphered data and applies its cryptographic material to retrieve the authentic user profile. At this point, the access-control method grants or revokes the reader depending on whether it has been authorized by the user running the privacy-preserving system or not.

3.4 Proposed architecture

The general structure of the proposed solution is depicted in Figure 1. Next, the main parts of the proposed architecture are briefly described.

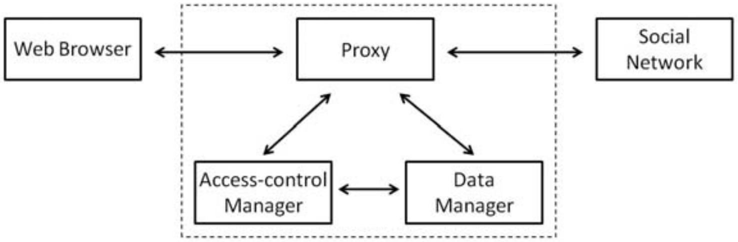

Fig. 1. Structure of the application

- *Proxy.* This module is the core of the application. Its target is to capture the HTML requests and responses that are transmitted between the *Web Browser* and the *Social Network* server (i.e., Facebook) and modify them in order to show the real data to the user in a transparent way. This implies that the user is not aware whether the real data is directly obtained from the user profile stored in the SNS or from the stego-object published in the Photo Album.
- *Data Manager.* The objective of this module is to manage which data is published in the SNS and which one is really shown when browsing *Facebook*. When a user wants to protect her profile, this module generates a stego-object and uses the *Proxy* module to publish it. On the other hand, when a user is browsing the protected profile of another individual, this module obtains the real information from the stego-object published in the Photo Album and submits this data to the *Proxy* module which is in charge of showing it to the user.
- *Access-control Manager.* This module manages the cryptographic material required to allow users to retrieve protected information. It also encrypts or decrypts information under request of the *Data Manager* module.

4 Our scheme in detail

In this section, we detail the two main algorithms that are used to protect personal data and retrieve it. After that, we focus on the steganographic technique used to hide the protected information in the SNS and also the method used to perform the cryptographic key management which is essential to perform a proper access control on the protected data. Finally, some deployability issues are discussed.

4.1 Proposed algorithms

The proposed system is formed by two main algorithms that focus on protecting the sensitive data of the user profile and retrieving it. The first procedure is executed by the user who wants to protect her privacy from any other entity of the system (e.g., other users, the SNS, external third parties, etc). The second procedure is run by any authorized user who wants to retrieve the protected information. Both are detailed in the following two subsections.

Algorithm-1: Protecting personal data. First, let us consider that a user profile P in a SNS is mainly a finite set of items I that provide some kind of information. This is $P = \{I_1, \dots, I_n\}$. Now, let us assume a user U_i who wants to protect some items of her user profile P_{U_i}. In order to do that, U_i executes the following protocol:

1. U_i requests to the SNS the web page that contains her user profile.
2. For each item I_w that U_i desires to keep private, she replaces the existent data with fake information. Note that, if the user is not comfortable with introducing fake data and the item is not mandatory for the SNS (e.g., birthday and gender are mandatory fields in Facebook), it is also possible to leave it blank.
3. U_i selects which users from her list of friends will be authorized to access the protected data.
4. The proposed system builds a XML file M that contains all the real data that must be protected. This file is then encrypted using the AES cryptosystem and the corresponding cryptographic key K_{U_i}. This is $C = E_{K_{U_i}}(M)$.
5. The system uses *broadcast encryption techniques* to perform the access control to the protected content according to the selection done by U_i in the step-3. In this way, key K_{U_i}, which is required to decrypt M, is ciphered according to the selected broadcast encryption technique (see Section 4.3 for more details about this). Let us denote the resulting element as λ. Note that the use of broadcasting encryption requires U_i to share a set of secret keys with each one of her friends in the SNS.
6. The system uses *steganographic techniques* (see Section 4.2 for more details about this) to hide C and λ in a *cover image* δ provided by U_i.
7. Finally, the system publishes the stego-object δ in the place reserved by the SNS to store images uploaded by users (i.e., Photo Album).

Algorithm-2: Retrieving protected data. Let us consider a user U_j who is also using the proposed privacy-preserving scheme. This user is browsing the Facebook profile of U_i and wants to retrieve a certain item I_w from P_{U_i}. In order to achieve that, U_j executes the following protocol:

1. U_j requests to the SNS the web page that contains the user profile of U_i.
2. The privacy-preserving system tries to find a valid stego-object δ in the place reserved by the SNS to store images uploaded by U_i (i.e., Photo Album). If δ is not found, it means that all the information published in the user profile is real and, hence, no further works is required and the protocol ends at this step. On the other hand, if δ is found, the proposed system continues the protocol.
3. The system uses a *steganographic method* (see Section 4.2 for more details about this) to obtain two items from the δ: (i) ciphertext C; and (ii) the access control element λ that was generated using a *broadcast encryption* method (see Section 4.3 for more details about this) and that contains a ciphered version of K_{U_i}.
4. The system uses the set of secret keys shared between U_j and U_i to retrieve K_{U_i} from λ. If U_j has not been authorized by U_i, U_j will retrieve an invalid key and she will be unable to get the real user profile of U_i. In other case, U_j obtains K_{U_i}, she is able to decrypt C and, hence, she gets the real content M (i.e., $D_{K_{U_i}}(C) = D_{K_{U_i}}(E_{K_{U_i}}(M)) = M$).
5. The system shows the real content to U_j instead of the fake information that is stored in the SNS. All the information is transparently shown to U_j using her own browser.

4.2 Hiding information from the SNS

As explained previously, SNSs generally do not allow their users to publish fake information in their accounts. Therefore, published fake data must look real in front of the SNS and the protected information must be hidden somewhere. In this way, the authors in [22] proposed to store all the protected data steganographed within images published in the SNS itself.

Using certain steganographic methods, a lot of data can be hidden inside standard images. Unfortunately, in this scenario, achieving a good *information rate* is not enough. More specifically, we require a steganographic scheme that also provides imperceptibility and robustness. Moreover, it should be oblivious (the recovery algorithm should not require the original unmarked image).

The well-known *F5* algorithm [23] fulfills the aforementioned requirements, hence, we first used it to hide information in the images uploaded to Facebook. Nevertheless, Facebook applies a heavy compression on the uploaded images, modifies the points-per-inch (ppi) to 72 ppi and changes any embedded profile to sRGB. As a result of all these transformations, no embedded data can be recovered from uploaded images marked with F5.

In order to overcome these difficulties, we developed a new stenographic algorithm robust enough to resist all the modifications currently applied by

Facebook. This algorithm is not the main contribution of this paper and, for this reason, we only give a brief description:

- *Embedding process.* First, the cover image is divided in cells of 8x8 pixels. Then, each cell is analyzed. If a cell is homogeneous (all pixels are similar), one bit of information is embedded, otherwise it is discarded. Finally, for each selected cell, we do the following: if we want to embed a "1", the less significant bits of each pixel are replaced with a certain fixed pattern a; otherwise, if we want to embed a "0", these bits are replaced with a certain fixed pattern b. Additionally, Reed-Solomon correcting codes [24] are used to improve the robustness.
- *Recovering process.* First, cells containing embedded information are identified. Then, for each one we get "0" or "1" depending on the number of pixels which are closer to pattern b or to pattern a. Finally, the correcting-codes retrieve the hidden information.

We have tested this method and it has been able to recover the embedded information from images uploaded to Facebook. Note that it is not the purpose of this paper to study the suitability of other stenographic algorithms present in the literature.

4.3 Access control and key management

A practical privacy-preserving scheme should not rely on a central server or require the users to be always on-line. In order to fulfill those requirements, we propose the use of *broadcast encryption* because it allows the owner of the protected data to grant or revoke access to one or several users in an easily way. Additionally, the owner can be off-line (i.e., users who try to get the protected information do not need to establish a direct connection with the owner). Instead of that, all the required access control data can be found embedded in the stego-object, together with the protected information.

In our implementation, we have used the well-known *Subset Difference (SD)* broadcast encryption scheme [25]. The reason is that it is is a particularly efficient scheme that generally requires a small amount of access control data even if there are several revoked users. Note that studying the deployability of other broadcast encryption schemes is out of the scope of this paper.

Finally, it is worth to mention that every user in our system uses a back-office application to interact with the "Access-control Manager". This application allows them to generate cryptographic keys for their friends and deny/grant access to their protected information. Then, these keys can be sent/received by e-mail. The list of friends and their email addresses can be directly found in the SNS.

4.4 Deployability issues

Even though the general idea of the proposed system can be applied to any SNS that offers classic functionalities (such as image uploading support, user profiles, etc), the implementation is completely platform-dependent due to the specific particularities of the HTML traffic generated by each SNS. This issue is not limited to the deployment of the proposed mechanism in different SNS, in fact, if the Facebook implementation changes, the *Proxy* module already implemented should be adapted to deal with the changes. This implies that a realistic privacy-preserving scheme based in our proposal should be continuously supported (e.g. by the open source community) in order to work properly. This shortcoming is shared with the scheme presented in [22].

5 Evaluation

In order to evaluate the performance of the proposed system we have measured the runtime cost of the following tests:

- *Test-1.* Retrieve the "User Profile" web page of a certain Facebook user U_i using a clean Firefox browser.
- *Test-2.* Retrieve the "User Profile" web page of U_i using a Firefox browser that is connected to Facebook through the *Proxy module* proposed in this paper. Note that, in this test, only the *Proxy* module is used.
- *Test-3.* Protect the "User Profile" of U_i using the proposed system.
- *Test-4.* An authorized user retrieves the *protected* "User Profile" web page of U_i.
- *Test-5.* A revoked user tries to get the *protected* "User Profile" web page of U_i.

All these tests have been run using a computer equipped with an Intel Core i7 at 2.7 Ghz, 8GByte of RAM, Windows 7 and DSL connection 10Mbit/1Mbit. Table 1 shows the different runtimes (in seconds) achieved by each test. The results provided are the average of 100 executions.

Table 1. Runtime cost (in seconds) for each test.

Test	Runtime cost
Test-1	4.886
Test-2	5.495
Test-3	4.137
Test-4	6.903
Test-5	5.638

It is worth to mention that these results represent the time required to fully download a *complete* "User Profile" web page. This point is relevant because, in addition to the requested profile, a "User Profile" web page also contains additional data such as advertisements, Facebook chat, etc. This fact justifies the 4.886 seconds required by Test-1.

Focusing on the time cost needed to obtain a protected user profile (Test-4), the overhead introduced by the proposed privacy-preserving scheme is around 2.017 seconds. We believe that this cost can be affordable for those users interested in explicitly controlling who can retrieve their personal data. Also, it is worth to mention that this is a first prototype and, probably, there is room for improvement.

6 Concluding remarks

In this paper, we have proposed a new system that enables the users of SNSs to protect their personal data. More specifically, by means of our proposal, they can exactly decide which individuals can access to their published information. As a result, even the SNS that hosts the user data cannot obtain any protected information if this is not explicitly allowed by the user. In addition to that, the new scheme has been designed to work properly with well-known SNSs such as Facebook.

Our scheme has been implemented and tested. We believe that the runtime costs obtained are quite competitive when compared with a direct connection to Facebook. More specifically, the proposed system introduces an approximate overhead of 2 seconds.

Regarding future work, it would be interesting to try to protect other sensitive elements which are present in SNSs such as user publications in the timeline/wall, the list of friends, etc.

Disclaimer and acknowledgments

This work was partly supported by the European Commission under FP7 project Inter-Trust, by the Spanish Ministry of Science and Innovation (through projects eAEGIS TSI2007-65406-C03-01, CO-PRIVACY TIN2011-27076-C03-01, ARES-CONSOLIDER INGENIO 2010 CSD2007-00004, Audit Transparency Voting Process IPT-430000-2010-31, ICWT TIN2012-32757 and BallotNext IPT-2012-0603-430000) and by the Government of Catalonia (under grant 2009 SGR 1135).

References

1. McMillan, G.: Twitter reveals active user number, how many actually say something. In: Time - Techland. (September 2011)
2. Consumer Reports National Research Center: Annual state of the net survey 2010. Consumer Reports **75**(6) (2010)
3. Zhang, C., Sun, J., Zhu, X., Fang, Y.: Privacy and security for online social networks: Challenges and opportunities. IEEE Network **24**(4) (2010) 13–18
4. Staddon, J., Huffaker, D., Larking, B., Sedley, A.: Are privacy concerns a turn-off? engagement and privacy in social networks. In: Proc. of the Eighth Symposium on Usable Privacy and Security – SOUPS'12. (2012)

5. Zheleva, E., Getoor, L.: To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles. In: Proc. of the 18th international conference on World wide web — WWW'09. (2009) 531–540
6. Van Eecke, P., Truyens, M.: Privacy and social networks. Computer Law & Security Review **26**(5) (2010) 535–546
7. Bilton, B.: Price of facebook privacy? start clicking. In: The New York Times. (May 2010)
8. Wilson, D.: Users are worried about social network security and privacy. The Inquirer (Oct. 2011)
9. Crimes, S.: Twitter sells old tweets to marketers - should users be worried? The Inquirer (Mar. 2012)
10. Dhia, I., Abdessalem, T., Sozio, M.: Primates: a privacy management system for social networks. In: Proc. of the 21st ACM international conference on Information and knowledge management – CIKM'12. (2012) 2746–2748
11. Cheek, G., Shehab, M.: Privacy management for online social networks. In: Proc. of the 21st international conference companion on World Wide Web – WWW'12. (2012) 475–476
12. Aimeur, E.: Privacy management for online social networks. In: Proc. of the International Conference on Availability, Reliability, and Security 2010 – ARES'10. (2010) 172–179
13. Baden, R., Bender, A., Spring, N., Bhattacharjee, B.: Persona: an online social network with user-defined privacy. In: Proc. of the ACM SIGCOMM 2009 conference on Data communication – SIGCOMM'09. (2009) 135–146
14. Diaspora: http://joindiaspora.com (last accessed: 12/02/2013) (2013)
15. Vaughan-Nichols, S.: Diaspora: It's no facebook ... yet. Computerworld (Sep. 2010)
16. Cutillo, L., Molva, R., Strufe, T.: Safebook: A privacy-preserving online social network leveraging on real-life trust. IEEE Communications Magazine **47**(12) (2009) 94–101
17. Vu, L., Aberer, K., Buchegger, S., Datta, A.: Enabling secure secret sharing in distributed online social networks. In: Proc. of the Annual Computer Security Applications Conference 2009 – ACSAC'09. (2009) 419–428
18. Nilizadeh, S., Jahid, S., Mittal, P., Borisov, N., Kapadia, A.: Cachet: a decentralized architecture for privacy preserving social networking with caching. In: Proc. of the 8th international conference on Emerging networking experiments and technologies – CoNEXT'12. (2012) 337–348
19. Scoble, R.: Facebook disabled my account. In: Scobleizer. (January 2008)
20. Guha, S., Tang, K., Francis, P.: NOYB: Privacy in online social networks. In: Proc. of the first workshop on Online social networks. (2008)
21. Luo, W., Xie, Q., Hengartner, U.: Facecloak: an architecture for user privacy on social networking sites. In: Proc. of the 2009 International Conference on Computational Science and Engineering. (2009) 26–33
22. Conti, M., Hasani, A., Crispo, B.: Virtual private social networks. In: Proc. of the first ACM conference on Data and application security and privacy – CODASPY'11. (2011) 39–50
23. Westfeld, A.: F5 - a steganographic algorithm. In: Proc. of the 4th International Workshop on Information Hiding - IHW'01. (2001) 289–302
24. Reed, I., Solomon, G.: Polynomial codes over certain finite fields. Journal of the Society for Industrial and Applied Mathematics (SIAM) **8**(2) (1960) 300304
25. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Proc. of Advances in Cryptology - CRYPTO'01. (2001) 4162