Privacy Protection in Social Networking Services

Daoyuan Li
Aalto University School of Science and Technology
daoyuan.li@aalto.fi

Abstract

As social networking services become increasingly popular, more and more attacks against users' private information are reported. As a result, privacy protection becomes an important concern among users. Previous research has produced many different approaches to deal with privacy control in different social networking sites. In this paper, we make a survey on different approaches proposed to tackle the privacy issue in social networking sites. In particular, we put current approaches into three general categories, i.e. approaches addressing end users' active participation, security automation based on machine learning algorithms, and privacy preserving by using a decentralized architecture for social networking services. Then we introduce and analyze some of the approaches in each category. Finally, we give some suggestions that may help privacy control in online social networks.

1 Introduction

Social networking services have been gaining popularity during the last few years. In social networking sites, users are encouraged to create their own profiles, write notes, upload pictures or videos, and join virtual social networks. These services are generally open to every Internet user and easy to access. Their openness has attracted many users. The largest social networking site, Facebook [1], has more than 500 million active users[2]. But the openness also comes with problems: malicious behaviors against service users are possible. For instance, people may get one another's private information, such as age, home address, mobile phone number and private pictures, even if for the user the information is not supposed to be exposed publicly.

There are several different strategies to control the exposure of private information. The most popular approach is to let users maintain a set of privacy rules, according to which a decision is made whether a certain user is able to view certain items. For example, Facebook let users customize privacy settings so that certain information can be accessed by some users or groups. However, these approaches are often not sufficient enough to protect users' privacy. They are either rather coarse-grained, or require a thorough understanding of the privacy control system and a huge amount of time and energy as well.

Fortunately, other approaches are proposed to protect users' private information. In this paper, we make a survey of various methods of protecting privacy in social networking services. We group them into three categories: methods focusing on end user participation, privacy policy making aided by machine learning methods and service decentralization. We examine how these approaches work and analyze their advantages and disadvantages.

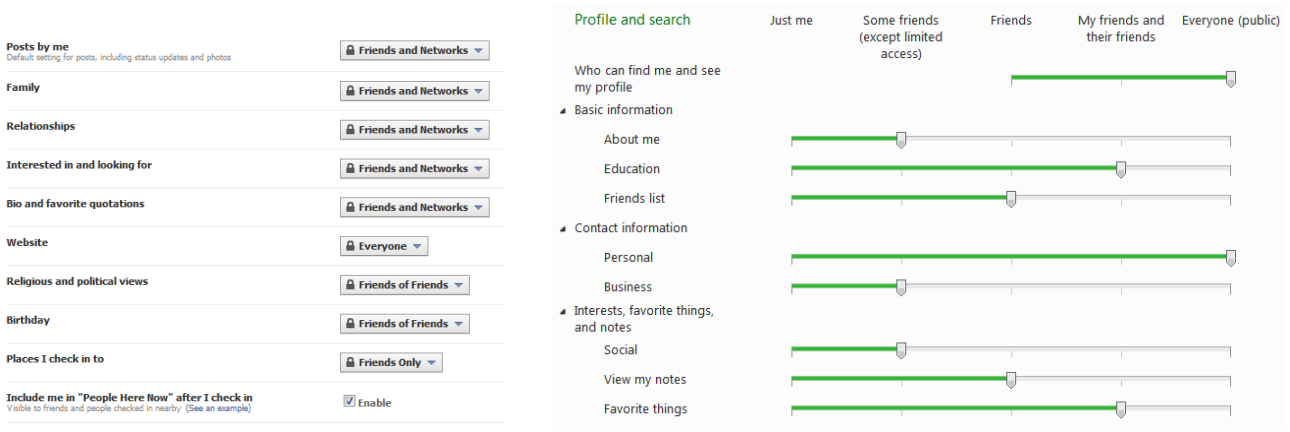
We give a brief introduction of how current privacy control approaches work and potential problems with them in Section 2. Then we discuss approaches that require users' active participation in Section 3. In Section 4, we look at how to use machine learning methods to help users making privacy policies. In Section 5 we will have a look at how peer to peer social networking architectures and their impact on social networking privacy control.

2 Background

The traditional way of protecting private information is to let users specify a set of rules, according to which decisions are made to allow or deny another user's access to certain items. For example, in Facebook, users can customize their privacy policies, so that only certain users and groups are allowed to view their information. Figure 1 shows parts of Facebook and Windows Live[3] Network's privacy customization pages. Users may allow a certain group of people's access to their different information. Facebook also allows users to keep *Block Lists*, in order to deny certain users' access to their information.

However, we argue that current approaches are often not effective enough. First of all, these strategies are often rather coarse-grained. And change of one privacy setting may result in unwanted or unexpected behaviors. Users can limit their information to Friends, Networks, or make it public, which involves respectively hundreds, thousands, and millions of users being able to access the information. Although in some occasions it is possible to gain a relatively fine-grained privacy policy, it requires thorough understanding of the whole privacy control system and quite often it may take a huge amount of time and energy to make a good privacy policy. Furthermore, there are occasions where users have limited resources to process privacy settings. For instance, mobile phone users may find it disturbing when many people send him/her request to see his/her profile, since he/she has a small screen full of different requests. This often undermines the usability of social network services. Moreover, end users are often the weakest link in the security chain [16]. They may have difficulties to understand security and

[1] http://www.facebook.com/
[2] http://www.facebook.com/press/info.php?statistics
[3] http://www.live.com



(a) Part of Facebook's Privacy Setting Page

(b) Part of Windows Live Network's Privacy Setting Page

Figure 1: Privacy settings on Facebook and Windows Live Network.

privacy policies designed by experts. Some users may even disregard privacy policies, which may create problems not only for themselves, but also for other fellow users.

Furthermore, since different pieces of information in social networks are often interleaved, one may be able to get some private information about another by using machine learning algorithms. For example, it is possible to infer undisclosed private information about individuals, using released social networking data [14]. The authors of [14] use a modified Naive Bayes learning algorithm and achieve 80% prediction accuracy on average, based on friendship links. Another paper [13] shows that private information can be inferred via social relations. And the stronger relationships people have in a network, the higher inference accuracy can be achieved.

3 End User Participation

As we have discussed above, current approach of letting users maintain privacy policies is not sufficient enough. However, users need to be a part of privacy protection, and there are several reasons for this. First of all, users are the weakest link in the security chain [16], thus we need to train users so that they have a better understanding of the system as well as the importance of privacy protection. If we can raise the security level of this weakest link, we can raise the security level of the whole system [21]. Furthermore, users have their rights to control their information. They are the owners of the information and have the ultimate control over the information. Finally, unlike other systems, it is difficult to treat social networking services as black boxes, so that the policy making process is controlled and monitored by security experts or computer systems. Since users are always a part of the system, we have no other choices but to train them patiently.

Therefore we should have some mechanisms to help users with their privacy tasks, and many approaches have been brought up to do so. These approaches address users' active participation in privacy policy making process. And to attract users doing this, usability becomes a big concern.

In the following sub-sections we will discuss how usability is achieved, and how to encourage users to participate in privacy policy making.

3.1 Security Usability

In software systems that are not security critical, their usability often focuses on human-computer interaction or user-centered design. When it comes to security issues, things becomes different [21]. What seems intuitive may actually not be user friendly. For instance, as argued in [6], while relocating a file by dragging the icon from one folder to another may be more usable than typing `mv file_name /home/another_folder`, when it comes to file access control, a tick-box with various options does not necessarily work better than `chmod`. Since after understanding the concept of Linux permission control mechanism, using command line tools is more intuitive and effective. As shown in Figure 2, when one want to give execution permission to a file, `chmod +x filename` may be easier than right clicked on the file, select 'Properties', switch to the 'Permission' tab, and then click on the 'Execution' tickbox.

Figure 2: GUI based permission control in Linux

Therefore, privacy control in social networking sites is often very complicated. According to [2], Facebook presents users 61 privacy control options spreading on 7 different settings pages. LinkedIn also has 52 settings on 18 pages.

And an average general-purpose social networking site offers 19.2 privacy settings on 3.7 separate pages.

Furthermore, since social networking services inevitably suffer from the *secondary goal problem*, i.e. security is not the primary goal and users always prefer doing social networking to managing access control rules, privacy control becomes a heavy burden for users.

While security usability is still an open research topic, an interesting interaction style is proposed in [5] to let users construct a textual description of their privacy wishes. In this case, the interface is more clear to users and less diffusive. Another suggestion in [5] is to provide a testing and debugging functionality, so that users are able to preview and validate their privacy policies.

3.2 Privacy Policy Collaboration and Sharing

We can see from the discussion above that creating and maintaining privacy policies tend to be a major challenge for most users, and lack of thorough understanding of the privacy control system can lead to unwanted or unexpected leakage of private information. Furthermore, configuring privacy policies is both a time- and energy-consuming task.

To make privacy control easier, users can collaborate with each other when controlling privacy policies [19]. For example, when a user Alice uploads some pictures to the online social network, she can appoint some of her friends Bob and Carol to manage the access permissions. This will make privacy control more easy to users.

Another way to ease privacy policy making is to share policies [1]. Users are able to view trusted privacy policies of others, save a copy of one policy or make some changes to it and then apply the policy for themselves. In this scenario, expert users make privacy policies available for general users, and general users just need to choose one pre-defined privacy policy that suits them best. However, the main challenge of this approach is establishing trust. Although a preliminary trust network already exists in social networks, it it possible that users without fully understanding of a policy apply it for themselves. In this case the (possibly malicious) policy may spread very fast in the online social network.

4 Security Automation

While it is possible to achieve a better privacy control by offering users more user-friendly interfaces, training users to get a better understanding of privacy control, and involving users more in practising controlling private information, human users are still the weakest link in the security chain.

As as result, researchers have brought up other approaches to protect users' privacy by decreasing the role of end users in privacy policy making. Machine learning methods are used to help users with policy making. In this section we will discuss some of the security automation approaches.

With machine learning approaches it is possible to achieve a finer-grained privacy control than traditional ones. And the way to achieve this is to limit the visibility of users' information within a smaller subset of their contacts, instead of all the contacts or make the information totally public. Another advantage is that it is easier to maintain a more dynamic policy using machine learning approaches. With traditional approaches users need to trivially control the access permission for every interaction to gain a fine-grained policy, while with machine learning algorithms polices can be learned from the context of the interaction.

In the following sub-section we will examine some of currently available machine learning approaches.

4.1 Current Approaches

A context inferring approach is proposed in [9]. This process is based on the social network of existing users. Then new users can be assigned to contexts extracted from the previous step. Finally, privacy policies can be assigned to contexts so that users in the same context share similar privacy policies.

Density of a sub-graph is defined and used for extracting context in [9], and it is calculated by dividing the number of actual edges l belonging to the sub-graph, with the maximum number of different edges $n(n-1)/2$, where n is the number of vertices in the graph. Therefore the density d of a sub-graph is

$$d = \frac{l}{n(n-1)/2} \tag{1}$$

For instance, if there are ten vertices and 15 edges in a graph, the density of the graph is $\frac{15}{10\times 9/2} \approx 0.33$.

Furthermore, a (K, γ, δ)-group is also defined in [9]. Assume there is a graph A of density at least γ, and a sub-graph $B \subset A$. Then (K, γ, δ)-group denotes there are at least K vertices in B, and the density of B is at least δ. And the sub-graph B is a (K, γ, δ)-group that contains at least K vertices and has the density of at least δ. B is also referred as a context of A.

Using Equation 1, a (K, γ, δ)-group, i.e. a context, can be drawn from a graph by calculating the density of sub-graphs and dividing a graph with low density into sub-graphs with higher density. Then interactions or information can be assigned to certain contexts with no or minimal help from the user.

Another similar machine learning method, a *Privacy Wizard* approach [11] interactively asks users simple questions, so that the wizard can assign labels to friends in the *Friend List*. After this process of active learning, a classifier is constructed to predict privacy preferences, based on a set of feature selection and community extraction. Selection of a good set of features is important and affects the accuracy and efficiency of the algorithm. In [11], two main types of features are considered: features based on extracted communities (utilizing extracted community structure), and other profile information (utilizing the user's friends' profiles). And regards to community extraction, there are numerous algorithms available to detect communities in graphs [12].

4.2 Deficiencies

Although in theory security automation is beneficial, in practice it may be not a panacea to cure end-user security problems. As argued in [10], there are several inherent limitations of security automation based on human and social factors.

And many of these factors are security technology independent. For example, predefining security policy may result in failures because of overly rigid specifications. Moreover, security decisions are inherently social when it comes to access control for human principals.

The inference approach seems feasible, however, we contend that although the calculation of sub-graph density is straightforward, it is not accurate enough. The reason is that the *Friend List* is often the source of privacy leakage. It is possible that users have people they do not actually know in their *Friend List*. For example, Sophos[4] conducted a probe[5] into how easy it is to steal user identities via Facebook in December, 2009. They created two false accounts and sent out 100 friend requests to randomly-chosen Facebook users from each account. And according to their result, within two weeks 95 users befriended (at least) one of the false users. And according to the result of a similar probe Sophos did two years before that, 41% users were willing to reveal their personal data to potential identity thieves[6].

Besides this, there are more reasons that we consider *Friend List* or *Connection List* is insufficient for protecting users private data. In social networking services networks are reflections of real-world relationships and networking circles. While in the real-world relationships between people are not reciprocated [18], in the undirected graph approach relationships are treated symmetrically. As a result this model does not accurately describe real-life situations. The model may either overstate or understate the degree of relationship intimacy. For example, let us assume there are two (to some extend) related persons, the president of one international corporation and an average employee in this company. Assume that the president does not know anything about the employee (which may be common in big companies), while the employee knows fairly enough information about the president. When we are modeling the relationship between these two, we come into a dilemma that whether the two should be related.

Furthermore, in the real-world relationships tend to change over time. Only very few relationships, such as biological relationships, may remain the same over a long period. Therefore having a not updated “friend list" does not necessarily ensure privacy, since people remains in the list until the user removes them out. But the problem is that there are no such mechanisms to remind users about potential risks. Hence it is highly possible that a friend one user added to the list long time ago would leak the user’s private information, intentionally or unintentionally.

Moreover, in case policies inferred by automated algorithms are not accurate or not suitable for the context, users should be able to adjust his/her own policies. As a result, policies should be visible and controllable to users. When requested, simple and comprehensible explanation should be offered to concerned users.

Last but not least, in most of current models relations are treated with equal degree of intimacy, while in the real-world this is not the case. In real-life we tend to have different intimacy level with different acquaintances. For instance, one person may be related and in contact with hundreds or even thousands of others, however his/her relationship need not, or could impossibly be the same with different people. He/She always have some close friends and others are just plain acquaintances. Therefore it is inaccurate to model all relationships with the same value in social networking services.

[4]http://www.sophos.com/

[5]http://www.sophos.com/pressoffice/news/articles/2009/12/facebook.html

[6]http://www.sophos.com/pressoffice/news/articles/2007/08/facebook.html

5 Service Decentralization

Besides the possibility of privacy leakage due to users’ poor privacy policies, it is also possible that service providers may leak out users information even if users have rather rigid policies. This section looks at some approaches to prevent service providers from potentially misusing or disclosing users’ private information.

A system named PeerSoN [4] was proposed as a distributed, peer-to-peer solution to fight against the service provider side problems. PeerSoN utilizes encryption and a public key infrastructure (PKI) to replace the centralized authority of classical social networking services. The chain of trust is built by exchanging keys among friends, and other friends’ credential authenticity are vouched by former trusted friends using the PKI. This system is also able to mitigate impersonation by implementing a challenge-response protocol.

Data exchange between users are done directly, instead of exchanging information via the centralized authority. First of all, the data being exchanged is encrypted, in order to prevent eavesdropping. Then the user gets another friend’s location from a look-up service. At last the user sends the data to the friend directly. In case the data receiving side is off line, the data is then stored in the look-up service and redistributed to the targeting user once he/she comes online.

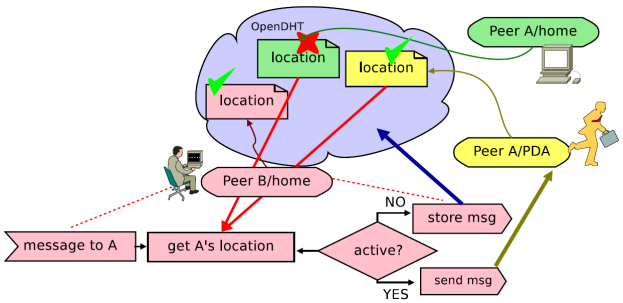


Figure 3: An example of message exchange in PeerSoN

PeerSoN uses Distributed Hash Table (DHT) as the look-up service. Figure 3 from [4] shows how one user B sends message to peer A. When the message is stored in the look-up service, the service provider is not able to view the content of the data, because the data is encrypted with B’s private key.

More similar peer-to-peer or decentralized approaches are also available in [3, 8]. These kinds of systems seem to work from the aspect of privacy control in social networking services. However, the usability of these systems are still unclear, since users have to really understand how to build up a chain of trust first. In the case of PeerSoN, users must know

how to use PKIs before they are able to use the peer-to-peer service.

6 Possible Improvements

As we have discussed in previous sections, neither encouragement of users' active participation in privacy policy making nor utilizing of security automation alone solves the security problems in social networking services. But proper use of machine learning techniques can actually result in more user-friendly policy options, and users' participation in policy making is more or less needed. Therefore in the rest of this section we give some suggestions that may improve the usability or effectiveness of privacy control in social networking sites.

6.1 Focus on Users' Behaviors

Since users are the owners of most data in online social networks, to protect their privacy we have to focus on their actual behaviors in social networks. Especially we should take users' communication with others and how privacy control is handled among them. It is intuitive that users contact people with similar background, experiences, interests and social networks. As a result, we can use machine learning methods to extract what and whom a certain user is interested in and then refine their privacy settings.

6.2 Smaller Communities

According to [17], users contact only with a very small subset of friends in their *Friend List*. For example, an average Facebook user with 50 friends contacts only 6 of his/her friends regularly, and only 4 of the friends have reciprocal communication with the user; a user with 150 friend contacts 9 friends and only 5 of the friends reply back; even a user with 500 friends contacts only 20 peers and barely 13 of them reply back.

Therefore, this gives us the impression that "friends" in *Friend List* are not all real friends, or at least not the type of friends we regularly make contact and share information with. Based on this observation, we propose a smaller community in online social networks.

For instance, if we keep a list of one user's most contacted friends, we can give suggestions to this user when he/she wants to share certain information. We may ask the user to only select a subset of his/her friends based on the frequency of communications between the user and friends.

In this way we are able to restrict users' private information within a smaller community, from the first step of users' information sharing.

6.3 More Dynamic Access Lists

In social networking sites such as Facebook and Linkedin[7], users have to maintain a list of acquaintances or some black lists to control who are able to view their information. And once one user Alice gets into Bob's white list, Alice is likely to stay in the list before she causes certain damages to Bob and got noticed by Bob. Otherwise Alice will remain in the list and is always able to observe what Bob is doing.

To fight against the deficiency mentioned above, we need a more dynamic access control mechanism. This mechanism should make sure that there are no unnecessary items in access lists and the lists should be maintained regularly, not just when the user receives a "request" from someone that wants to add the user to his/her friend list. We can do this by regularly reminding users to check their access lists, and providing users detailed information about potential risks when keeping someone in the lists for a long time while never contacting him/her.

6.4 Default Settings

Research has shown that most users do not change default settings [15]. As a result a default privacy settings are of great importance for privacy control. However, according to Facebook's privacy policy[8], the default privacy setting for certain types of information you post on Facebook is set to "everyone", including users' status, photos, notes, family and relationship status.

And another poll [7] shows that a significant majority of users would like to have the opportunity of "opt-in" rather than "opt-out". This suggests that a more rigid default privacy control policy should be available for users, it would then improve both security and usability.

6.5 Consistent User Interfaces

We have discussed that it is often inherently complicated for users to fully understand security related rules and policies. Therefore by providing an ever-changing privacy setting interface would not result in users' better understanding. Instead, with a consistent user interface users are able to take in new concepts faster.

6.6 Interactive Settings

Instead of overwhelming users with full pages of privacy settings, we propose an interactive way of making privacy policies, based on the study result that users are more willing to provide feedbacks and are more likely to benefit from richer interactions [20].

Users should also be able to view what their privacy settings will produce. Facebook already has an option for users to preview their profiles from the perspective of " most people on Facebook". Facebook users users are also able to see themselves from the perspective of any friends. It gives users a more vivid impression of what their privacy policies do exactly.

7 Conclusion

In this paper we categorized these approaches into three groups: methods addressing user participation, focusing on security automation and use of a decentralized service. Then

[7] http://www.linkedin.com/

[8] http://www.facebook.com/policy.php

we briefly introduced some of the methods in each category and analyzed their key features and main deficiencies. Finally we gave some suggestions that may help privacy control in online social networks. We may improve the privacy control condition by focusing on users' actual behaviors, creating smaller communities, use of more dynamic access control lists, and so on. Building a privacy-violation free online social networking service is no easy task, especially when users are not able to fully understand what they are doing exactly. Still, it is possible to improve the systems and mechanisms, so that less violations of privacy may be made.

References

[1] J. Bonneau, J. Anderson, and L. Church. Privacy suites: Shared privacy for social networks. In *Workshop on Social Networks, SIGCOMM*, 2009.

[2] J. Bonneau and S. Preibusch. The privacy jungle: On the market for data protection in social networks. In *The Eighth Workshop on the Economics of Information Security (WEIS 2009)*, 2009.

[3] S. Buchegger and A. Datta. A case for P2P infrastructure for social networks - opportunities and challenges. In *Proceedings of WONS 2009, The Sixth International Conference on Wireless On-demand Network Systems and Services*, Snowbird, Utah, USA, February 2-4, 2009.

[4] S. Buchegger, D. Schiöberg, L. H. Vu, and A. Datta. Peerson: P2p social networking: early experiences and insights. In *SNS '09: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, pages 46–52, New York, NY, USA, 2009. ACM.

[5] L. Church, J. Anderson, J. Bonneau, and F. Stajano. Privacy stories: Confidence in privacy behaviors through end user programming. In *In SOUPS 2009: Symposium On Usable Privacy and Security, Mountain View*, 2009.

[6] L. Church and A. Whitten. Generative usability: security and user centered design beyond the appliance. In *NSPW '09: Proceedings of the 2009 workshop on New security paradigms workshop*, pages 51–58, New York, NY, USA, 2009. ACM.

[7] G. Cluley. Poll: 93% say facebook should make you 'opt-in' to sharing rather than 'opt-out'. http://www.sophos.com/blogs/gc/g/2010/05/27/poll-93-facebook-optin-sharing-optout/, 2010.

[8] L. A. Cutillo, R. Molva, and T. Strufe. Privacy preserving social networking through decentralization. In *2009 Sixth International Conference on Wireless On-Demand Network Systems and Services*, pages 145–152. IEEE, February 2009.

[9] G. Danezis. Inferring privacy policies for social networking services. In *AISec '09: Proceedings of the 2nd ACM workshop on Security and artificial intelligence*, pages 5–10, New York, NY, USA, 2009. ACM.

[10] W. K. Edwards, E. S. Poole, and J. Stoll. Security automation considered harmful? In *NSPW '07: Proceedings of the 2007 Workshop on New Security Paradigms*, pages 33–42, New York, NY, USA, 2008. ACM.

[11] L. Fang and K. LeFevre. Privacy wizards for social networking sites. In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 351–360, New York, NY, USA, April 2010. ACM.

[12] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.

[13] J. He, W. Chu, and Z. Liu. Inferring privacy information from social networks. In S. Mehrotra, D. D. Zeng, H. Chen, B. Thuraisingham, and F.-Y. Wang, editors, *Intelligence and Security Informatics*, volume 3975 of *Lecture Notes in Computer Science*, chapter 14, pages 154–165. Springer-Verlag, Berlin/Heidelberg, 2006.

[14] J. Lindamood, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. Inferring private information using social network data. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 1145–1146, New York, NY, USA, 2009. ACM.

[15] W. E. Mackay. Triggers and barriers to customizing software. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 153–160, New York, NY, USA, 1991. ACM.

[16] K. D. Mitnick and W. L. Simon. *The Art of Deception: Controlling the Human Element of Security*. John Wiley & Sons, Inc., New York, NY, USA, 2002.

[17] S. Sandberg. How many friends can you have? http://blog.facebook.com/blog.php?post=72975227130, 2009.

[18] J. Scott. *Social network analysis: A handbook*. Sage, 2000.

[19] A. C. Squicciarini, M. Shehab, and F. Paci. Collective privacy management in social networks. In *Proceedings of the IEEE International Symposium on Policies for Distributed Systems and Networks (POLICY)*, Richmond, VA., 2010.

[20] S. Stumpf, V. Rajaram, L. Li, M. Burnett, T. Dietterich, E. Sullivan, R. Drummond, and J. Herlocker. Toward harnessing user feedback for machine learning. In *IUI '07: Proceedings of the 12th international conference on Intelligent user interfaces*, New York, NY, USA, 2007. ACM.

[21] A. Whitten. *Making Security Usable*. PhD thesis, Carnegie Mellon University, 2004.