Exploiting Privacy Policy Conflicts in Online Social Networks

Akira Yamada, Tiffany Hyun-Jin Kim, and Adrian Perrig

February 23, 2012

CMU-CyLab-12-005

CyLab
Carnegie Mellon University
Pittsburgh, PA 15213

Exploiting Privacy Policy Conflicts in Online Social Networks

Akira Yamada
KDDI R&D Laboratories Inc.
2-1-15 Ohara Fujimino
Saitama, 356–08502 Japan
yamada.akira@kddilabs.jp

Tiffany Hyun-Jin Kim
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
hyunjin1@ece.cmu.edu

Adrian Perrig
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
adrian@ece.cmu.edu

ABSTRACT

Online Social Networks (OSNs) offer access control mechanisms to protect users' sensitive information from undesired accesses. Yet, their information is still vulnerable to disclosure when their friends assign *conflicting* privacy policies: a user prohibits everyone from accessing his own content or profile but his friends allow others to see it. OSNs tend to select *Permit-Take-Precedence* when resolving multiple conflicting policies so that the information is possibly exposed regardless of the information owner's preference. In this paper, we confirm that specific types of information in real OSN services are under this circumstance. We then propose three attacking scenarios that reveal the hidden friend-lists, profiles, and posted messages on users' OSN accounts, exploiting a target's sensitive information. We finally discuss possible countermeasures in terms of both implementation and human behavior.

1. INTRODUCTION

Online Social Networks (OSNs) have become popular web sites, which enable users to communicate with their friends easier than offline social activities. Indeed, Facebook has announced that the number of active users surpassed 750 million as of July 6, 2011[1] and its traffic has surpassed that of the search engine giant Google in the U.S. in 2010.[2] OSNs are convenient; particularly younger generations seem to be addicted to OSNs [7]. However, new privacy issues are emerging [7, 15, 1, 31, 35, 30] as summarized below.

Although Online Social Network (OSN) platforms offer access control mechanisms to prevent undesired access, many researchers have been raising security and privacy concerns. Unlike traditional access control models, OSNs possess unique issues: non-expert users need to determine their own privacy policies rather than just posting or browsing content, situations surrounding users dynamically change by users' sending and accepting friend requests, and most information is shared or owned by multiple users. Users thereby have to count on the communities' and/or their friends' capabilities to control the shared information [15, 1]. However, individual users can assign mutually conflicting privacy settings.

More specifically, when Alice posts a message on Bob's message board, both Alice and Bob have a chance to determine the publication setting. Alice may want to keep her message private between Alice and Bob, but Bob can publish his board after she posts. Note that any information that is shared among friends, a group, or a community with inconsistent settings is plagued by this problem. The fundamental reason is that OSNs' policy combining procedure is inconsistent, and it tends to choose *Permit-Take-Precedence* policy (positive authorizations prevail over negative ones) rather than *Deny-Take-Precedence* policy (negative authorizations prevail over positive ones) when resolving users' privacy policy conflicts.

In the access control research field, the policy conflict detection and resolution have been hot topics [20, 23, 36, 33]. However, most researches focused on traditional Role-Based Access Control (RBAC) systems where security administrators configure the policies. Recently some researchers proposed an automatic policy conflict resolution in OSNs [13, 2, 12], but they did not examine specific OSNs' functionality and their policy combining procedures. Researchers also exploited specific services with previously proposed attacks [16, 37, 18, 32, 26, 3, 4, 28]. However, such exploits required collecting an enormous number of users' data and/or simply inferring hidden information rather than explicitly exposing it.

In this paper, we examine these privacy policy conflicts in the context of multiple OSN services (e.g., Facebook, Google+, etc.), and indicate that specific types of information are vulnerable to privacy breaches. We then propose three attack scenarios (friend-list recovery, profile recovery, and post recovery attacks) to expose hidden information, and evaluate their effectiveness in the context of Facebook. As a result, we confirm that the proposed attacks can successfully recover a target user's sensitive information with reasonable web requests. Our contributions are as follows:

- We confirm that the privacy policy conflicts exist in real OSN services. Real OSNs' policy combining processes are not consistent, and OSNs tend to choose *Permit-Take-Precedence* and leak sensitive information. We list specific cases of policy conflicts in 5 popular OSNs.
- We propose 3 attack scenarios, Friend-list/Profile/Wall Post recovery attacks, which expose a specific user's personal information based on the discovered policy conflicts. These attacks are more efficient and more accurate than previously discovered attacks when targeting a specific victim user.
- We conducted a user study to analyze how much novice users were concerned about these policy conflicts and how they would react if these conflicts are discovered. Our

[1] `http://www.facebook.com/press/info.php?statistics`
[2] `http://www.startribune.com/business/112694054.html`

Table 1: Information types with privacy conflicts in popular OSNs.

Info. Type	Description
`friend`	Friend-lists are shared among users.
`stream`	Posts on message boards are shared with the owners and the other users who post messages.
`comment`	Comments on message boards.
`group_member`	Groups' membership information is shared among their members.
`like`	Items are shared with users who are commonly interested in.
`photo_tag`	Photo tags are shared with the owners of the photos and the tagged users.
`video_tag`	Video tags are shared in the same manner as photo tags.

Table 2: Comparison of the privacy policy conflicts in different OSNs.

	friend	stream	comment	like	group _member	photo video _tag
Facebook	+	+	+	+	+	–
MySpace	+	–	+	+	*	+
Orkut	++	–	+	+	+	+
Twitter	+	–	*	–	+	*
Google+	+	–	+	–	*	–

+: Sensitive information can leak, ++: No privacy setting available (always leaks), –: Sensitive information cannot leak, *: Not implemented.

study results show that more than half of the study participants were willing to change their privacy policies to resolve policy conflicts and to protect the privacy of other users.

- We discuss possible countermeasures to avoid these policy conflicts in OSNs in terms of both implementation and human behavior.

2. PRIVACY POLICY CONFLICTS

A privacy policy conflict occurs when more than two users co-own an identical piece of information, and both of them are capable of configuring their own publication settings individually. Note that no leakage happens if all of the related users' settings are consistent. In case of a message board system, two potential users exist: who owns the board and who submits a message, and both can choose their own publication settings.

Policy Conflicts in OSNs. We discover that 7 types of information cause policy conflicts in popular OSNs as shown in Table 1: `friend`, `stream` (posts on message board), `comment`, `group_member`, `like`, `photo_tag` and `video_tag` (for simplicity, we use the widely adapted Facebook's terminologies[3]). In case of a friend-list, the platform has a `friend` table that contains all pairs of friends. If an ith user U_i configures his privacy setting to publish his friend-list, the platform returns his friend list F_i to any user's query. However, an entry $\langle U_i, U_j \rangle$ in a friend-list F_i is related to not only the list owner U_i but also all of U_i's friends. This implies that a part of U_j's friend relationships is also revealed *regardless of U_j's privacy configuration.* As a result, if U_j's friends publish their friend-lists, an adversary can reconstruct U_j's private friend-list F_j.

Comparison with popular OSNs. Table 2 compares the privacy policy conflicts over popular OSNs: Facebook, Myspace, Orkut, Twitter, and Google+ as of September 15, 2011. Note that OSNs are continuously updating their functionality and privacy setting. Facebook provides the most fine-grained access control policies to preserve users' privacy, but it can potentially leak users' sensitive information as we show in this paper. Recently, Facebook fixed the privacy setting of photo tag,[4] following its risk suggested by the researchers.[5] In Facebook and Orkut, users can post a message on other users' message board, but Orkut offers different access policy where both the owner of the message board and the authors can control the setting individually. However, the owner cannot grab the entire control of the message board without assistance from all authors. Unlike Facebook and MySpace, Orkut does not offer any settings to preserve users' friend-lists (i.e., anyone can collect them). Twitter and Google+ adapt directional friend-lists such that a user has two independent friend-lists: *followers* and *followings*. As a result, they can define independent publication policies for their own friend-lists. However, these services allow users to display not only their *followings* but also *followers*, which causes a policy conflict.

3. ANALYSIS

In this section, we evaluate the privacy settings of OSN users in the context of Facebook and describe three attack scenarios that can expose hidden information.

3.1 Privacy Settings of Facebook Users

We conducted multiple experiments on Facebook to confirm the impact of the privacy policy conflicts. As an initial step before launching any attack, we collected users' configurations on their Facebook pages to check how much they opened their data to be publicly accessible. We developed a web crawler by combining web APIs and web scraping techniques. We started with a user's friend-list as an initial seed, and crawled up to 2-hop friends (i.e., friends of friends).

Table 3 summarizes Facebook users' configurations, where 64.3% of users shared their friend-lists, and 47.8% of users shared the preferences in their profile pages. We also observed that only 7.8% of users opened their walls to public. This implies that a user cannot rely on the group or the community to protect his sensitive information because of the difference in the settings. For example, 35.7% of users did not share their friend-lists, but this crawling revealed at least one friend of them. Though most users do not share their walls, their messages are possibly exposed to everyone with 7.8% chance, if users post messages on the friends' without being aware of the owners' settings.

[3] `http://developers.facebook.com/docs/reference/fql/`

[4] `https://blog.facebook.com/blog.php?post=10150251867797131`

[5] `http://www.heinz.cmu.edu/~acquisti/face-recognition-study-FAQ/`

Table 3: Users privacy settings in Facebook.

Configuration	Share		Not share	
Connect	140,224	(93.6%)	9,638	(6.4%)
Profile	18,927	(12.6%)	130,938	(87.4%)
Education	52,855	(35.3%)	97,007	(64.7%)
Interests	71,565	(47.8%)	78,297	(52.2%)
Friends	96,337	(64.3%)	53,525	(35.7%)
Notes	14,043	(9.4%)	135,819	(90.6%)
Feed	11,660	(7.8%)	138,202	(92.2%)
Photos	1,514	(1.0%)	148,348	(99.0%)
Total	149,862			

Connect indicates that users can receive friend requests from any other users. Profile indicates users' basic information except education and interests.

We retrieved all public friend-lists, profiles and wall posts of the above collected users. An average user had 549.5 friends in the friend-list, and expressed 11.3 items of interest on the profile page. There were 105.13 posts and comments on an average wall. 7,000 (10%) users did not show any item of interest in the profile pages and 850 (0.8%) users had no message on their walls.

3.2 Definition of OSNs' parameters

We define OSN's parameters used in the attack scenarios. When a user starts using an OSN, he creates a *profile* page to indicate his background, such as name, sex, affiliations, photos, and interests. He can upload his contents on his own *message board* which includes a stream of posts. He can also utilize the *friend-list* which contains friends of the user to define the perimeter so that unknown users outside this perimeter cannot access his personal information.

Profile page. We define users in an OSN as U_i $(0 \leq i < n)$ where n is the number of users. Each user U_i owns a profile page A_i, a friend-list F_i, and a message board W_i. A profile page contains a user's personal preference, and we define a set of groups as G_k $(0 \leq k < n_k)$ where n_k is the number of groups. G_k represents a group of users with similar taste, interest, or affiliations. In the context of a general OSN model, we do not consider individual attributes, such as age, sex, hometown, etc. When U_i becomes a member of a group G_k $(G_k \leftarrow G_k \cup U_i)$, U_i's profile A_i includes the index of the group.

$$A_i = \{\text{index}(G_{k_i}) \mid U_i \in G_{k_i}\}$$

Friend-list.[6] The friend relationships in an OSN can be denoted as an undirected graph $G = (V, E)$, where vertex V represent OSN members and edges E between two vertice represent the friend relationship $\langle U_i, U_j \rangle$, where $F(U_i, U_j)$ is a function that returns True (T) if U_i and U_j are friends, and False (F) otherwise:

$$\begin{aligned} V &= \{U_i \mid 0 \leq i < N\} \\ E &= \{\langle U_i, U_j \rangle \mid \forall i, j \in [0, n-1], F(U_i, U_j) = T\}. \end{aligned}$$

U_i's friend-list F_i is a subset of V who are friends with U_i. When U_i joins an OSN for the first time, his friend-list is initially empty: $F_i = \emptyset$. When U_i befriends other members (i.e., U_i sends a friend request to U_j and U_j accepts it), both U_i's and U_j's friend-lists are updated as follows: $F_i \leftarrow F_i \cup U_j$, and $F_j \leftarrow F_j \cup U_i$.

$$F_i = \{U_j \mid \forall i, j \in [0, n-1], F(U_i, U_j) = T\}.$$

Message board. Each user U_i owns a message board W_i. On a message board, a user can post any content he wishes to share with other users. The x^{th} post $W_i[x]$ on a board W_i consists of an author user $W_i[x].User$ and a corresponding message $W_i[x].Message$:

$$W_i[x] = \langle W_i[x].User, W_i[x].Message \rangle.$$

Multiple types of contents can be posted (e.g., messages, photos, videos), but we do not distinguish them. Some OSNs also distinguish posts from comments. U_i, his friends, and any members of the OSN can potentially post some content on U_i's board W_i.

3.3 Attack Scenarios

We randomly selected 107 users who did not share their friend-lists, profile pages, and walls. Note that we can obtain at least one of the friends for each target because we recursively crawl publicly-available friend-lists. We now describe three attack scenarios that can expose hidden information on any OSN platforms with the policy conflicts. We compared the result of these attacks with the retrieved information in Section 3.1 to calculate accuracy of these attacks.

Friend-list recovery attack. This attack enables an adversary to recover a target's friend-list that the target explicitly configures to protect. An adversary initially discovers some of the target's friends, and recursively gathers their friend-lists to successively recover the target's friend-list.

Algorithm 1 presents the crawling algorithm. The parameter S stands for initial seeds of a victim's friends $U_s (0 \leq s < n_s)$ where n_s is the number of seeds. To select the next crawling users, an adversary confirms whether these users have friend relationships with the victim using the function OldREST API `friends.areFriends` in Facebook. After confirming the friend relationships, the adversary retrieves the friends' friend-lists by the web scraping function `scrape.friends()`, which returns an array of friends. Eventually, this algorithm outputs the victim's undisclosed friend-list F'_t. The web API may not exist on some other OSNs, but the attacker can still launch this attack without the web API.

Figures 4 (a) illustrates the cumulative density functions comparing public and revealed private information. We recovered 434.37 friends for a user on average, and there was only one case for whom we could not extract more than one friend due to the failure of discovering sufficient initial seeds. Although we could not directly verify the actual number of friends, the revealed private information was about 79% of the friends in the public friend-lists. The web API `friends.areFriends` provides friend relationships whether two users are friends or not, if one of the users share the friend-list. The upper bound of this attack depends on the distribution of users' privacy settings.

Profile Recovery Attack. This attack discloses a user's preferences in his profile page. The adversary first collects a victim's friend-list, which we assume to be accessible, for example by launching the friend-list recovery attack. The adversary then utilizes web scraping for collecting the profiles of the victim's friends. The `group_member` and `like`

[6]This general model does not include Twitter's or Google+'s style of directional friend-lists.

Input: U_t //Target user ID
$S = \{U_s | 0 \leq s < o\}$ //Seed user IDs
Output: F'_t //Recovered friend-list
1: $F'_t \leftarrow ()$//Initialize
2: **repeat**
3: $F \leftarrow ()$//Initialize no-crawled friends
4: **for all** $U_s \in S$ **do**
5: **if** friends.areFriends$(U_s, U_t) = T$ **and** $U_s \notin F'_t$ **then**
6: $F \leftarrow F \cup U_s$
7: $F'_t \leftarrow F'_t \cup U_s$
8: **end if**
9: **end for**
10: $FF \leftarrow ()$//Initialize friends of friends
11: $S \leftarrow ()$
12: **for all** $U_f \in F$ **do**
13: $FF \leftarrow$ scrape.friends(U_f)
14: $S \leftarrow S \cup FF$
15: **end for**
16: **until** $S = \emptyset$
17: **return** F'_t

Figure 1: Friend-list recovery algorithm.

Input: U_t //Target user ID
Output: A'_t //Recovered profile
1: $A'_t \leftarrow ()$ //Initialize
2: $F_t \leftarrow ()$ //Initialize friend-list
3: $F_t \leftarrow$ scrape.friends(U_t)
4: **for all** $U_f \in F_t$ **do**
5: $A_f \leftarrow ()$//Initialize friend's attributes
6: $A_f \leftarrow$ scrape.profile(U_f)
7: **for all** $G_f \in A_f$ **do**
8: **if** page.isFan$(G_f, U_t) = T$ **then**
9: $A'_t \leftarrow index(G_f) \cup A'_t$
10: **end if**
11: **end for**
12: **end for**
13: **return** A'_t

Figure 2: Profile recovery algorithm.

Input: U_t //Target user ID
Output: W'_t //Recovered posts
1: $W'_t \leftarrow ()$ //Initialize
2: $F_t \leftarrow ()$ //Initialize friend-list
3: $F_t \leftarrow$ scrape.friends(U_t)
4: $W_f \leftarrow ()$ //Initialize friend's wall
5: **for all** $U_f \in F_t$ **do**
6: $W_f \leftarrow$ stream.get(U_f)
7: **for all** $W_f[x] \in W_f$ **do**
8: **if** $W_f[x].User = U_t$ **then**
9: $W'_t \leftarrow W_f[x] \cup W'_t$
10: **end if**
11: **end for**
12: **end for**
13: **return** W'_t

Figure 3: Posts recovery algorithm.

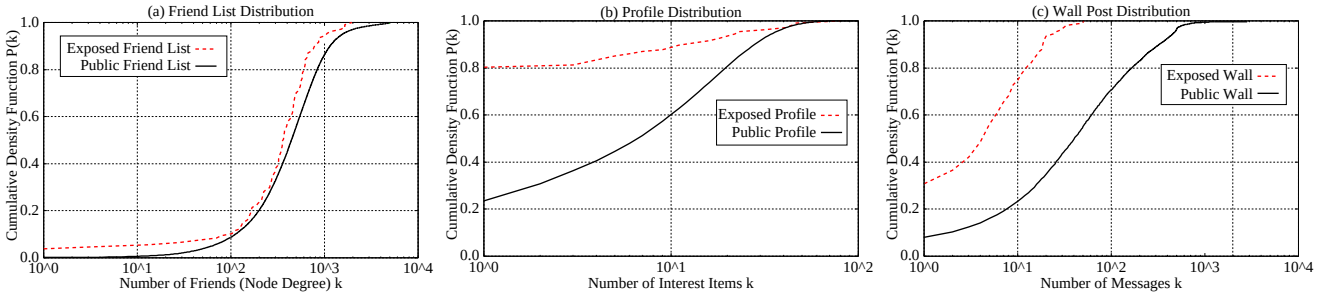

Figure 4: Revealed information by the recovery attacks.

can potentially leak sensitive information so that the adversary can identify members of the groups and users who are interested in the objects.

Algorithm 2 shows the profile recovery. This algorithm outputs the recovered profile page A'_t in the target U_t's profile. The function `scrape.profile()` corresponds to a web scraping of a given user's profile, and returns a list of preferences. The function `pages.isFan()` returns whether a user's profile includes the preference page or not, and corresponds to an OldREST API `pages.isFan`. Though this API does not exist in other OSN platforms, an adversary can look up the members of a group if it is vulnerable to the friend-side-channel leak.

Figure 4 (b) illustrates the density function of the recovered profiles. The attack extracted at least one item of interest from 20.6% of targets with average 3.98 items, which is about 35% of the public profiles. Compared to the previous attack, we were unable to recover as many target's items. However, since a user may be interested in any items not limited to friends' items, we can increase the number of recovered items by checking not only friends' items of interest but also popular items.

Post Recovery Attack. This attack exposes a victim's posts that he publishes on his friends' walls. While an adversary cannot access the victim's wall when he does not share it with everyone, the adversary can possibly collect the victim's posts or comments from his friends' walls. The victim has no control over the posts on a wall of someone else, even though he is the author of the posts.

We illustrate the algorithm for the post recovery attack in Algorithm 3. An adversary collects posts $W_f[x]$ from a victim U_t who protects his own wall while leaving posts and comments on other users' walls W_f $(f \neq t)$. The function `stream.get()` corresponds to an OldREST API `stream.get`, and returns all the posts on a given user U_f's wall. This algorithm outputs the recovered posts W'_t of victim U_t. If such API is not available in the OSN platform, the attacker can simply utilize web scraping to obtain the list of posts.

Figure 4 (c) shows the exposed messages by the authors who did not share their walls but posted on public walls. This attack discovered 7.6 hidden messages per user on average, and revealed at least one message from 79.4% of the targeted users. We can infer that these messages might be

accidentally or carelessly posted on public walls.

3.4 Attack Efficacy

In our exploits, we select next crawling users adaptively. Hence, we do not have to gather huge number of other users' profile to recover a given target's sensitive information. For example, Asuncion et al.'s attack [3] requires crawling an entire group that a target possibly belongs in. Additionally, our attacks exploit privacy conflicts to *e*xplicitly reveal sensitive information rather than implicitly as other inference attacks do [37, 24, 26, 16]. Most of previous work is based on machine learning algorithms or probabilistic estimations, which can produce candidates of secret information with probabilities but cannot identify the covert information explicitly.

4. EVALUATION

In the previous sections. we have presented that most OSNs possess the friend-side-channel problem where users' privacy policies possibly conflict with other's policies. Since every user has an individual priority, it is impossible to change each privacy policy to eliminate all conflicts comprehensively. However, if users can observe and convey each other's policies, they can mitigate conflicts by fixing their policies. To explore such acceptable solutions, we conducted an online user study by using Amazon Mechanical Turk (MTurk). We designed the questionnaire following guidelines [19, 11] to filter out people who answer carelessly or arbitrarily. We recruited 155 Facebook users who live in the United States by setting the location restriction flag on MTurk. We carefully eliminated 57 unqualified participants who were unable to answer obviously verifiable questions.

Demography. 36.7% identified themselves as males and 63.3% identified themselves as females. In terms of age, 35.7% were in the age range of 18–24, 40.8% were in 25–34, 16.3% were in 35–45, and 6.2% were younger or older than the above ranges. The majority of people were familiar with computers. In terms of computer usage, they reported to use a computer for more than 2 hours per day: 27.6% use for 2–4 hours, 37.8% use for 4–8 hours, and 28.6% use for 8–16 hours. Over half participants were heavy Facebook users who spent more than 1 hour on Facebook per day: 28.4% for 1–2 hours, 18.7% for 2–4 hours, and 9.7% for 4–8 hours. 92.9% of them created the Facebook account more than 1 year ago; 24.5% created the account 1–2 years ago, 26.5% created the account 2–3 years ago, 17.3% created the account 3–4 years ago, and 24.5% created the account more than 4 years ago. Here are sample questions that we asked:

- How sensitive do you think profile, friend-list, messages on wall, photo, and video are?
- Do you share profile, friend-list, messages on wall, photo, and video?
- Are you concerned about the friend-side-channel leak in OSNs?
- How much are you willing to ask your friends to change their privacy settings so that they do not leak any of your information?
- Are you willing to change your privacy configuration so that you do not leak any of your friends' information?

User activity in Facebook. The participants had certain amount of personal information in Facebook. In term of profile, most had their correct profile in Facebook, but some of them did not fill correct information or left it blank; 93.4% input their real name, 91.8% filled their gender, 69.4% filled the age, 68.4% filled their country, 69.4% filled their education, and 33.7% filled their company information. In terms of their daily activities on Facebook, they reported to post messages, photos, and/or videos on Facebook as shown in Figure 5 (a). The graph summarizes the amount of participants' personal information in Facebook. 68.4% of users have more than 100 friends, and 61.2% have more than 100 posts on their walls. 43.9% have more than 100 photos in Facebook, but none of them actively upload their videos.

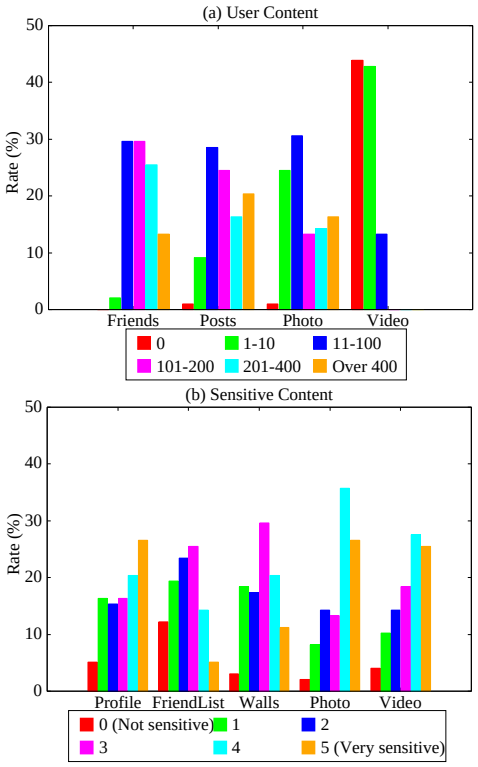

Figure 5: Users' content and users' attitudes toward the content.

Privacy concern. The participants were concern with the privacy in OSNs and regarded most of their personal information in OSNs as sensitive. We first asked participants how much they are concerned with privacy in 6 degrees (1 is low and 6 is high). None of them chose an option 1 and 84.7% of participants chose more than degree 4. We then inquired how much they regard specific types of information as sensitive. Figure 5 (b) illustrates the participants responses for each type. A group of people (26.5%) considered that their profile is particularly sensitive. Users regarded videos as the most sensitive, and photos, messages, and friend-list as the second, third, and fourth sensitive, respectively.

To confirm users' privacy behaviors, we provided some specific cases where they could possibly disclose personal information without noticing: when they sign up for discount coupons and shopping membership cards. While most users were concerned with privacy, 40.8% of them provided an email address or a phone number to receive coupons,

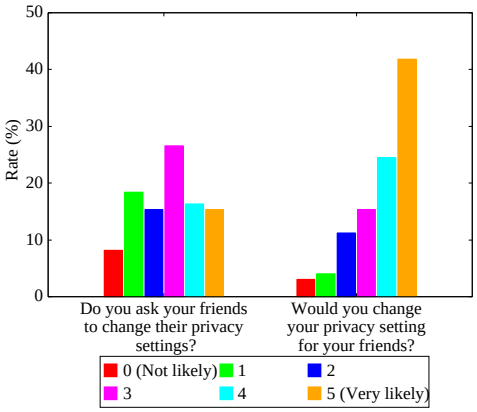

Figure 6: Privacy setting negotiation questions.

and 77.6% of users reported to have signed up for shopping membership cards. This result suggests that there are many opportunities where users casually sacrifice their personal information.

Privacy setting. More than half of participants limited access to their personal information from unknown users. Users choose *Share* only when they share the type of information with everyone, and *Not share* otherwise. 31.6% of the participants published their friend-lists and 15.3% of them set their walls to be publicly visible. 13.3% and 12.2% of users shared their photos and videos with everyone. Compared to the previous analysis in Table 3, which is from web crawling, more users reported to share their photos or videos with everyone. A possible reason is that the crawler assumes *Not share* if no information is visible in the profile pages, and some of participants have no photo in Facebook.

Recognition of the friend-side-channel. Most users do not know the friend-size-channel before our survey, and are concerned about the leakage of the channel. We explain the participants about the mechanism of access control in OSNs and the friend-side-channel leak. 77.6% of participants do not know the problem before our survey, and 63.3% of participants report that they are concerned about this friend-side-channel problem.

Policy negotiation. We ask them if they are willing to change their privacy settings to prevent the channel leaks and if they ask their friends to update the friends' settings to prevent from the leakage. Figure 6 summarizes the answers where most participants (81.6%) are willing to change their setting, and more than half (58.1%) of them ask their friends to update their configurations. Interestingly, the participants change passively more than to ask the friends actively.

Summary of user study. This user study suggests that most users are concerned with privacy in general, but some can casually scarify their personal information. Malicious services can exploit users' sensitive information by leveraging the friend-side-channel leak. Some users cannot understand specific constrains that their sensitive information becomes vulnerable. Particularly, it is difficult to let non-expert people to understand the access control mechanism and the relationship between their privacy settings and the consequences. In terms of privacy setting, most users are willing to change their own policies in order to help their friends. Half of users actively request their friends to update the privacy setting after learning about this threat. This implies that users can change their privacy settings if they are requested by their friends or if a tangible evidence of a security breach is provided to them.

5. DISCUSSION

In this section, we discuss countermeasures and future works for preventing the conflicting privacy configurations.

Challenge of privacy policy conflicts. As long as users can choose individual priorities, these conflicts are inevitable. Even in a close group of *friends of friends*, members have disparate preferences. According to our preliminary experiments in Table 3, we discovered that OSNs' users have different preferences and the policy conflicts indeed exist. In other words, the current privacy settings in OSNs cannot satisfy all users' requirements. We believe that this problem is related not only to technological but also to ethical and philosophical aspect. For example, if you have a celebrity as your friend, you may want to share the friend-list even though this friend is not willing to. Because OSNs are relatively recent services in our society, prior works do not investigate how to resolve these conflicts in the virtual world.

Deny-Take-Precedence. A simple solution to resolve the conflicts is to modify the behavior of OSN platforms to prevent any information leakage. For example, the current OSN platforms allow a user to share his friend-list even though his friends do not want to share theirs. Instead, the OSN platforms can employ *Deny-Take-Precedence* to prevent the disclosure of any undesired private information while combining multiple users' privacy policies. Although such a filtering mechanism may not harmonize all users' preferences, it eliminates the undesirable privacy breaches. This precedence may cause slower online communications because the rate of finding other users and establishing connections with them may decrease.

Leak-proof OSN design. With a different approach, dividing a conflicting table into two separate tables can become an alternative countermeasure. As we present in the comparison of OSNs in Table 2, Twitter and Google+ provides 2 separate friend-lists: *following* and *followers*. In this case, the owner can only maintain the *following* table, and when user Alice adds Bob as a new friend, Bob's user ID is entered to Alice's *following* friend-list, and Bob notices that Alice is now one of his *followers*. Twitter and Google+ thereby implement conflict-free friend-lists if they do not display *followers* in the profile pages. One limitation is that adversaries can still infer friends of a target from such tables. Because two separate tables are not independent from each other, one possibly contains others' records, and an adversary can infer entries of a hidden table from the public tables of a victim's friends. Despite such a limitation, maintaining separate friend-lists will minimize the privacy breach of users.

Privacy policy negotiation. Some prior work explores automatic policy negotiation systems [10, 29, 17]. For ex-

ample, Cresenzo et al. propose an automatically evolving access control [10], which configures policies based on given requirements of utility and privacy. Squicciarini et al. apply the game theory to resolve the collective enforcement of the privacy policy on the shared data [29]. However, such automatic negotiations cannot satisfy all users, and possibly cause side effects. In the game theory-based approach, for example, users cannot protect their content explicitly if others sharing the content strongly request to publish it.

Nevertheless, we believe that automatic policy negotiations are heading toward an ideal solution. It is an analogy with real human relationships where people choose friends or communities with similar preferences. For instance, if you are a first-grade student of a university, you may be willing to find new friends while sacrificing your privacy. In such a case, you can increase the opportunities to encounter new people by making your profile search-able by others who are also looking for new friends. After several months or years, you may have sufficient number of friends, at which point you no longer need to publish your profile on public, and the same may hold for your friends. You can then change your setting and ask your friends to switch their settings. In case some of your friends are unwilling to change their privacy settings, you may disconnect such *unreliable* friend links to protect not only your own but also the rest of your reliable friends' privacy. Consequently, users who have similar privacy preferences may converge into communities and mitigate the policy conflicts.

6. RELATED WORK

address formalizing the access control model of OSNs [13, 2, 12]; however, none of them evaluate behaviors of popular OSN services. Fong et al. try to formalize the dynamics of Facebook-style OSNs where users send and accept friend requests each other [13], but they do not take co-owned information account. Some papers address the policy conflict detection and resolution in general [20, 23, 36, 33], but they focus on general RBAC systems rather than OSNs. Ni et al. [27] and Glenn and Huth [8] propose to apply the multi-valued logic, which produces not only boolean "grant" or "deny" but also "undefined" or "conflict." While these models can express a situation of "conflict," they cannot resolve conflicts by themselves.

Some prior work explores attacks to disclose personal information in OSNs, but these attacks are inefficient or inaccurate. For example, some require gathering an enormous number of users and/or inferring hidden information rather than explicitly exposing it [16, 24, 37, 18, 32, 26, 3, 4, 28]. Heatherly and Lindamood propose an inference attack by using Naïve Bayes classifier [16, 24], and Mislove et al. target the attributes of community such as high school, university, employer, interests and so on [26]. Asuncion et al. focus on recovering friend links [3], but this attack requires a huge amount of data for their group testing algorithm.

Some research groups [21, 6, 5] analyze the effects of personal information leakage using an easier constraint where an attacker can compromise or subvert users' accounts. Our attack scenarios are more challenging as we do not rely on compromising user accounts. Although Bonneau et al. imply the possibility of an attack combining crawling and compromising, they do not analyze feasibility to recover the friend-list without compromising accounts [6]. An attacker can create fake user accounts using public information and requests other users to accept the friend request [5].

Balduzzi et al. and Polakis et al. propose an attack to recover email addresses by crawling and querying multiple OSNs [4, 28]. By guessing email addresses from user names, they generate millions of candidate mail addresses and confirm the relationship with OSN accounts. Some other research groups compare multiple OSNs and identify the personal attributes [18, 32].

Several researchers work on web crawling in OSNs [9, 25, 22, 34, 14], which may be related to this paper, but they do not exploit a specific target with limited computational resources; they collect entire users. While some researchers propose a strategy to choose the next node of crawling, they focus on unbiased sampling [25], which generates a subset of friend-relationship graphs with similar properties to the entire graph.

7. CONCLUSION

Although OSNs attempt to improve security and privacy, they have not achieved the complete or ideal access control mechanisms that users actually demand. In current OSNs, individual users can choose different preferences, causing privacy conflicts in shared information that multiple users co-own. In this paper, we examine real OSN services in terms of their policy combining and resolution processes, and we confirm that they tend to choose *Permit-Take-Precedence* when resolving conflicting policies. We also indicate specific cases that OSNs leak users' sensitive information regardless of the owners' preference due to these conflicts. We then propose 3 recovery attacks that abuse these conflicts to expose a specific user's personal information using less computational resources than previous attacks.

The problem of data co-ownership is one of the fundamental problems in modern OSNs, not only from a technical but also from an ethical, legal, and philosophical perspective. There is no obvious solution on how an OSN should handle two users with conflicting privacy settings. If a user wants to publish his friend-list, does he have the right to do so regardless of his friends' preferences? Who should decide? However, it is of paramount importance of assessing the consequences of shared information in the context of practical services so as to provide a criteria for users and operators to identify the ideal behavior. We believe that this paper presents a first step toward a future user-oriented privacy policy in OSNs.

8. REFERENCES

[1] R. Acquisti and R. Gross. Imagined communities: Awareness, information sharing, and privacy on the facebook. In *6th Workshop on Privacy Enhancing Technologies*, 2006.

[2] M. Anwar, Z. Zhao, and P. W. L. Fong. An access control model for facebook-style social network systems. Technical report, University of Calgary, 2010.

[3] A. U. Asuncion and M. T. Goodrich. Turning privacy leaks into floods: surreptitious discovery of social network friendships and other sensitive binary attribute vectors. In *9th Annual ACM Workshop on Privacy in the Electronic Society*, 2010.

[4] M. Balduzzi, C. Platzer, T. Holz, E. Kirda, D. Balzarotti, and C. Kruegel. Abusing social networks for automated user profiling. In *13th International Conference on Recent Advances in Intrusion Detection*, 2011.

[5] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identity theft attacks

on social networks. In *18th International Conference on World Wide Web*, 2009.

[6] J. Bonneau, J. Anderson, and G. Danezis. Prying data out of a social network. In *2009 International Conference on Advances in Social Network Analysis and Mining*, 2009.

[7] D. Boyd and D. Buckingham. Why Youth (Heart) Social Network Sites: The Role of Networked Publics in Teenage Social Life. In *Youth, Identity, and Digital Media*. 2008.

[8] G. Bruns and M. Huth. Access-control policies via belnap logic: Effective and efficient composition and analysis. In *Proceedings of the 2008 21st IEEE Computer Security Foundations Symposium*, 2008.

[9] D. H. Chau, S. Pandit, S. Wang, and C. Faloutsos. Parallel crawling for online social networks. In *16th International Conference on World Wide Web*, 2007.

[10] G. Crescenzo and R. J. Lipton. Social network privacy via evolving access control. In *4th International Conference on Wireless Algorithms, Systems, and Applications*, 2009.

[11] J. S. Downs, M. B. Holbrook, S. Sheng, and L. F. Cranor. Are your participants gaming the system?: screening mechanical turk workers. In *28th international conference on Human factors in computing systems*, 2010.

[12] P. W. Fong. Relationship-based access control: protection model and policy language. In *Proceedings of the first ACM conference on Data and application security and privacy*, 2011.

[13] P. W. L. Fong, M. Anwar, and Z. Zhao. A privacy preservation model for facebook-style social network systems. In *14th European Conference on Research in Computer Security*, 2009.

[14] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou. Walking in Facebook: A case study of unbiased sampling of OSNs. In *IEEE INFOCOM 2010*, 2010.

[15] R. Gross and A. Acquisti. Information revelation and privacy in online social networks. In *2005 ACM Workshop on Privacy in the Electronic Society*, 2005.

[16] R. Heatherly, M. Kantarcioglu, B. Thuraisingham, and J. Lindamood. Preventing private information inference attacks on social networks. Technical report, The University of Texas at Dallas, 2009.

[17] H. Hu and G.-J. Ahn. Multiparty Authorization Framework for Data Sharing in Online Social Networks. In *Data and Applications Security and Privacy*. 2011.

[18] D. Irani, S. Webb, K. Li, and C. Pu. Large online social footprints–an emerging threat. In *2009 International Conference on Computational Science and Engineering*, 2009.

[19] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, 2008.

[20] M. Koch, L. V. Mancini, and F. Parisi-Presicce. Conflict detection and resolution in access control policy specifications. In *Proceedings of the 5th International Conference on Foundations of Software Science and Computation Structures*, 2002.

[21] A. Korolova, R. Motwani, S. U. Nabar, and Y. Xu. Link privacy in social networks. In *17th ACM Conference on Information and Knowledge Management*, 2008.

[22] B. Krishnamurthy and C. E. Wills. Characterizing privacy in online social networks. In *First Workshop on Online Social Networks*, 2008.

[23] N. Li, Q. Wang, W. Qardaji, E. Bertino, P. Rao, J. Lobo, and D. Lin. Access control policy combining: theory meets practice. In *Proceedings of the 14th ACM symposium on Access control models and technologies*, 2009.

[24] J. Lindamood, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. Inferring private information using social network data. In *18th International Conference on World Wide Web*, 2009.

[25] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *7th Internet Measurement Conference*, 2007.

[26] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel. You are who you know: inferring user profiles in online social networks. In *Third ACM International Conference on Web Search and Data Mining*, 2010.

[27] Q. Ni, E. Bertino, and J. Lobo. D-algebra for composing access control policy decisions. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, 2009.

[28] I. Polakis, G. Kontaxis, S. Antonatos, E. Gessiou, T. Petsas, and E. P. Markatos. Using social networks to harvest email addresses. In *9th annual ACM workshop on Privacy in the Electronic Society*, 2010.

[29] A. Squicciarini, M. Shehab, and J. Wede. Privacy policies for shared content in social network sites. *The VLDB Journal*, 2010.

[30] F. Stutzman, R. Capra, and J. Thompson. Factors mediating disclosure in social network sites. *Computer Humman Behavior*, 2011.

[31] Z. Tufekci. Can you see me now? audience and disclosure regulation in online social network sites. *Bulletin of Science, Technology & Society*, 2008.

[32] J. Vosecky, D. Hong, and V. Y. Shen. User identification across multiple social networks. In *First International Conference on Networked Digital Technologies*, 2009.

[33] B. Wu, X.-y. Chen, Y.-f. Zhang, and X.-d. Dai. An extensible intra access control policy conflict detection algorithm. In *Proceedings of the 2009 International Conference on Computational Intelligence and Security - Volume 01*, 2009.

[34] S. Ye, J. Lang, and F. Wu. Crawling online social graphs. In *Asia-Pacific Web Conference*, 2010.

[35] A. L. Young and A. Quan-Haase. Information revelation and internet privacy concerns on social network sites: a case study of facebook. In *Fourth International Conference on Communities and Technologies*, 2009.

[36] W. D. Yu and E. Nayak. An algorithmic approach to authorization rules conflict resolution in software security. In *Proceedings of the 2008 32nd Annual IEEE International Computer Software and Applications Conference*, 2008.

[37] E. Zheleva and L. Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *18th International Conference on World Wide Web*, 2009.