Received 21 March 2014
Revised 21 March 2014
Accepted 22 April 2014

Calculation of account reachability risk for users having multiple SNS accounts from user's profile and regional information

Ayano Yoshikuni
Department of Computer Science, Ochanomizu University, Tokyo, Japan, and
Chiemi Watanabe
Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Japan

Abstract

Purpose – The purpose of this paper was to argue that social networking service users should be vigilant in protecting the relationship between multiple accounts.

Design/methodology/approach – In this paper, the authors propose the use of account reachability, a measure of privacy risk which demonstrates the possibility of a stranger finding a user's private account based on information in their public account. In addition, they present ARChecker, a tool to calculate the value of account reachability. ARChecker also provides advice on how to modify the user's profiles and messages to decrease the privacy risk.

Findings – The system very simply checks account reachability and shows the result and reasons for accessibility to personal information. From the results, users can learn how to protect themselves from privacy risk by taking certain measures.

Originality/value – This paper proposes account reachability, a new measure of privacy risk, and presents ARChecker, a tool to calculate the value of account reachability.

Keywords Privacy, Web, ARChecker, Social networking service, Account reachability, Privacy risk

Paper type Research paper

1. Introduction

As online social networking services (SNSs) have become popular, the number of SNS users has increased. As of this writing, Facebook announced it had 1.11 billion active accounts and Twitter had over 500 million users. SNSs have become essential tools in our everyday life. We share our daily experiences with our friends on online SNSs by writing short messages, uploading photos and movies, etc.

However, most people are not concerned with other users' personal information. Even if a user reveals their home location, it is unlikely to trigger a dangerous incident,

International Journal of Web Information Systems
Vol. 11 No. 1, 2015
pp. 120-138
Emerald Group Publishing Limited
1744-0084
DOI 10.1108/IJWIS-03-2014-0010

This work was supported by JSPS KAKENHI Grant Number 24700088.

such as a stranger appearing at the front door. That said, it is important for users to bear in mind that they may easily become a victim of a cyberstalker over a trivial matter. Users should therefore pay careful attention to limit the information revealed to other users to avoid becoming a victim of cyberstalking.

We propose *account reachability checker (ARChecker),* a tool that shows the personal information accessible to a cyberstalker from a user's profiles and messages on SNSs (Yoshikuni and Watanabe, 2013). ARChecker focuses on users who have multiple accounts on different SNSs. As an example, we assume that a user has two accounts: one on Facebook to write messages as a technical blogger, and another on Twitter to write messages about his private life and communicate with his friends. He believes that only few friends know that he owns these two accounts.

Imagine a case in which a user has trouble with customers from an indiscreet comment he makes on Facebook, and the customers happen to find his Twitter account. They may collect his personal information such as his home address, girlfriend's name and any compromising pictures to cause him mental distress. In this way, revealing the user's private account through his public account could lead to serious repercussions.

In this paper, we define a measure named *account reachability (AR),* which calculates the possibility of revealing a private account through a public account. To calculate *AR,* we define an attack model, which uses profiles and messages of a public account to find the private account, and we next define a formula to calculate the AR. ARChecker calculates the *AR* from a Facebook account to a Twitter account based on the definition. It simulates a simple attack, which searches for Twitter accounts based on the profiles of the Facebook account, and calculates the *AR.* For an intuitive understanding of the result, ARChecker visualizes the result using icons. In addition, it shows which keywords affect the value of the *AR.*

The remainder of the paper is organized as follows. Section 2 describes the privacy risk of exposing a user's multiple accounts and related works. Section 3 describes our proposed tool named ARChecker, and defines the attack model, and the measure of privacy risk named AR. Section 4 discusses the effectiveness of our proposed tool and the privacy measure. Section 5 introduces the variations of the attack based on the defined attack model. Section 6 concludes the paper.

2. Privacy risk of exposure of a user's multiple accounts

2.1 Flow of information gathering

Cyberstalkers stalk or harass an individual, a group of individuals or an organization via the Internet, which includes voyeurism, making threats, identity theft and defamation (Zheleva and Getoor, 2011). Cyberstalking could result from a trivial matter. For example, a user may be a victim of a cyberstalker who falls in love with their picture, or an indiscreet message on an SNS may incur the wrath of many people, and cause the perpetrator to receive serious levels of harassment from strangers. If cyberstalkers are not related to their victim in real life, they may gather private information about them from the Web. Here, we introduce an actual cyberstalking incident that happened in Japan, to show the flow of gathering of the victim's private information from SNSs, and to demonstrate the reason we focus on cases involving victims with multiple SNS accounts.

This incident commenced on Twitter when the victim bragged about his immoral activities in his office. Enraged on finding his message, another user disseminated his

message to some SNSs. As punishment, many members of a Web forum started cyberstalking him to obtain his personal information, which in turn was exposed on one forum after another. In the end, he had no option but to quit his job.

Figure 1 shows the flow of how the victim's private information was collected:

- The victim did not reveal his personal information on Twitter, but he had another account on Mixi (Japanese social network service). He used his Mixi account for communicating with his friends in the real world.
- A stalker was able to locate his Mixi account because his Twitter account was in the same name as his Mixi account.
- The victim's address, office name and room layout of his house were extracted from his profiles on Mixi and subsequently exposed on a forum.
- The victim uploaded movies on his Mixi account, enabling the stalkers to expose a picture of his face on the forum.
- A stalker found the Web site of his rock band.
- The victim had not used his real name on Twitter, but the stalker found his name on the rock band's Web site.

As this case suggests, the cyberstalkers could glean little information from his messages or profile on his Twitter account; however, the victim became vulnerable to attack when the cyberstalkers identified his Mixi account, which revealed more of his personal information. The followers of his Twitter account were mainly cyberspace acquaintances, so he was careful not to write any personal information on his messages. On the other hand, his friends on his Mixi account were his real friends, so he was less vigilant about detailing his private experiences.

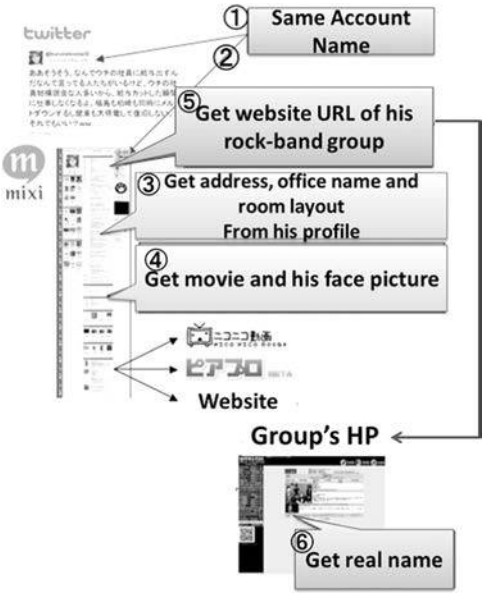

Figure 1.
Stream identifying individual data

Alex Patriquin of Compete.com reports that most users have accounts on several SNSs (Pulse Compete Pulse, 2007). A Japanese government survey reports that 60 per cent of users have multiple SNS accounts. As a *Forbes* reporter described in *Forbes* (2011), users who have multiple accounts use their accounts for different purposes, such as for sharing information with business associates or for communicating with friends from high school. Users generally write their profiles and messages according to their purpose of the account.

Therefore, users tend to write different information in the messages and profiles of different accounts. For example, a friend of the *Forbes* reporter uses Twitter to write articles related to her job, such as trend analysis of social marketing. On the other hand, she uses her Facebook account for communicating with her friends and family and she posts personal experiments. Therefore, aggregating private information from multiple accounts causes unintended personal information leakage, as described in Section 2.1.

However, it is difficult to manage different personalities on corresponding accounts over long periods. A user who always posts messages that are related to his work may post messages which offer glimpses into his daily life. In addition, it is a hard task to keep track of what personal information was told to whom. For example, imagine that a user shares his or her Facebook account with only real friends when starting on Facebook, in time the user makes friends with a person who is not close, or a person whom he or she cannot recall in detail. It is, therefore, important to verify that information shared on each account is different from other accounts.

2.2 Related works

The focus of this paper is on unintended personal information leakage from multiple SNS accounts. The following two cases describe situations in which cyberstalkers obtain a victim's private information from social network data:

(1) *Obtaining anonymized social network graph data*: The entire data of social networks are leveraged for data analysis such as marketing research, social research, advertisements and recommendations. Social network data are represented by a large graph, which expresses the relationship between accounts with nodes and edges. To protect a user's personal information, the data should be anonymized before publishing. However, some de-anonymization attacks have been reported (Narayanan and Shmatikov, 2009). If an attacker succeeds in his attack, the data will be used for collecting a victim's personal information. Several papers proposed privacy measures (Zheleva and Getoor, 2011) for social network data based on anonymity measures for microdata, such as k-anonymity (Sweeney, 2002) and l-diversity (Machanavajjhala *et al.*, 2007). In this case, the data owner such as the social network service provider anonymizes the data.

(2) *Crawled data via an API or screen scraping*: An attacker can collect information from the Web pages of online social networking services. In general, users specify their privacy settings that control the release of their profiles and messages. For example, Twitter provides a privacy setting option for users to release their messages and list of friends only to those followers authorized by the user. In this case, each user should self-manage his personal information and anonymize his account from unknown users; they should apply anonymization

techniques such as using pseudonyms and sanitizing their identifiable information.

Focusing on the latter case, users should self-manage their personal information by keeping track of who knows the information released in their messages and profiles; however, this is a heavy task for SNS users. Many papers have highlighted the risk of unconscious information leakage from social network data. Joinson (2008) and Boyd and Hargittai (2010) report that many SNS users do not specify privacy settings or change the default privacy settings. Akcora *et al.* (2012) describe how the creation of a relationship without specifying the appropriate privacy setting may expose the user to risk of unconscious release of personal data, and it proposes the risk model for social networks. Dubrawski *et al.* (2009), Carminati *et al.* (2011) and Wang *et al.* (2011) have also proposed the risk model to aid the users' understanding of potential privacy risks. Even if a user is vigilant in setting the privacy settings of the SNS, there remains a risk of an unconscious personal information leakage. Mislove *et al.* (2010) remarked that some data analysis methods based on friendship relations may infer the personal information of an account even if the account does not release any privacy information. Sadilek *et al.* (2012) has proposed a technique for inferring the user's areas of residence and work from his friends' action histories.

These papers deal with the privacy concerns of a single SNS. On the other hand, Irani *et al.* (2012) have modeled unintended personal information leakage from multiple online social networks. It mentions the risk of aggregating the victim's personal information from different accounts, and it models the privacy risk of aggregate attribute leakage. We also define the privacy risk of multiple SNS accounts. However, we focus on whether an attacker can locate the different accounts of victim. If the attacker cannot establish the sole ownership of two different accounts, the user can avoid an aggregate leakage attack.

3. Account reachability checker

Our proposed tool, ARChecker, checks whether a cyberstalker can find a secret SNS account from the information of another SNS account. In summary, it simulates the process for finding the target account and calculates the AR from the result of the attack. We define the *AR,* which is the possibility that a cyberstalker can find the secret SNS account from the information in the given SNS account. Section 3.1 shows the overview of ARChecker, while Section 3.2 defines the AR.

3.1 Overview

Figure 2 shows a screenshot image of the ARChecker. It shows the value of the AR, and for an intuitive understanding of the value, it visualizes the result using human icons (Figure 2③). If the user's icon is displayed much bigger or in a greater number than other human icons, there is a high possibility of finding the user's secret account. In addition, it shows the words that are included in the profiles or messages of messages of a given SNS account and gives advice for protecting the target account. The words are displayed using word clouds (Figure 2④). When the user removes the words listed in the clouds from the messages and profiles in a given account and they recheck the AR, the value should decrease. Through repetition of this process of calculating the AR and modifying the profiles (or messages), the possibility of cyberstalkers finding the user's secret account reduces.

Figure 2.
Interface of the account reachability checker

The current version of ARChecker supports Twitter and Facebook, that is it calculates the AR of a user's Twitter account from the information in the user's Facebook account.

Figure 3 shows the process flow for checking AR. In this figure, we assume that a user has a Facebook account (s_1) and Twitter account (s_2). They use s_1 for their work and s_2 for chatting with their real friends:

- The user authenticates their Facebook account (s_1) and Twitter account (s_2) with an *authentication module*.
- The *Profile collection module* obtains the user's profile data from account s_1.
- The *AR calculator module* calculates the AR from s_1 to s_2.
- The *Result Visualizer* shows the result.

3.2 Attack model

This section defines the attack model which is assumed in ARChecker. The purpose of the attack model is to locate the victim's secret account, s_2. The attacker already knows the victim's account, s_1. The relationship of the attacker's account with s_1 can be considered as one of the following:

- *A friend in the real world*: The attacker and s_1 know each other in real life.
- *A "friend" in the SNS*: The attacker has a "friend (or follower/followee)" relationship with s_1, but they do not have relationship in the real world.

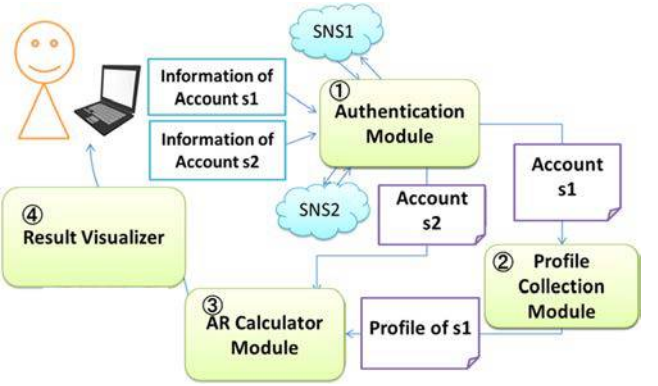

Figure 3.
Process flow for checking AR

- *No relationship*: The attacker does not have relationship with s_1 either in the real world or on an SNS.
- *Privileged account*: The attacker has a privileged account on an SNS, such as an administrator, allowing access to s_1's personal information including profiles, messages and access log.

In this paper, we deal with the second and third relationships. We model the processes of attack as in the following text.

3.2.1 Step 1: collecting candidates. An attacker searches for possible candidates of account s_2 by using the information in account s_1. Information about account s_1 is obtained through the attacker's relationship between s_1 and the attacker's account. We assume that the attacker searches for the candidates of s_2 by issuing queries in the SNS. To search for the candidates of s_2, they extract some keywords from messages and profiles of s_1. We define a set of queries Q for searching the candidates of s_2 as in the following text.

3.2.1.1 Definition 1 (queries for collecting candidates). Let s_1 be the account that is already known by the attacker, and let $s_1 . prof$ and $s_1 . msg$ be messages and profiles of s_1. The function $GenQueries(s_1 . prof, s_1 . msg)$ is implemented by the attacker to extract queries for searching for s_2. Let Q be a set of queries that are extracted by *GenQueries* as follows:

$$Q = \{q \mid q \in GenQueries(s_1 \cdot prof, s_1 \cdot msg)\}$$

3.2.1.2 Definition 2 (candidate account). Let $Cand(q)$ be a set of candidate accounts that are selected by query q.

3.2.2 Step 2: calculating similarities. The attacker calculates the similarities between each candidate account and s_1. If the attacker determines multiple candidates from Step 1, they can find s_2 from the highest similarity score. The attacker implements a calculation model based on the features of the accounts. In its simplest form, the attacker can count the number of attributes that have the same value in the two accounts. If s_2 and the candidates have many messages in the SNS, the attacker can calculate the similarity of writing style by using author estimation techniques (Stamatatos, 2009). The attacker can also use the time and location of the messages to calculate the similarities.

Section 3.4 introduces a simple implementation as an example, and Section 5 discusses advanced techniques for calculating the similarities of users.

3.2.2.1 Definition 3 (similarity score function). Let $Score(s_1, c)$ be a function that is implemented by an attacker to calculate the similarity between s_1 and a candidate c.

3.3 Account reachability

Based on the attack model described in the previous section, we define AR as in the following text.

3.3.1 Definition 4 (AR). Let s_1 and s_2 be accounts which are used by user u; attacker a already knows s_1 but they do not know s_2. AR is the possibility that a finds account s_2 from the information of account s_1.

Based on the process of the attack model, we define a formula for calculating AR as in the following text.

3.3.2 Definition 5. Assuming an attacker selects candidates of s_2 by using query q, that is $Cand(q)$. AR by q is calculated as:

$$AR(s_1 \rightarrow s_2, q) = Match(s_2, C\,and\,(q)) * \frac{Score(s_1, s_2)}{\Sigma_{c \in C\,and\,(q)} Score(s_1, c)}$$

$$Match(s_2) = \begin{cases} 1\ if\ s_2 \in C\,and\,(q) \\ 0\ else \end{cases}.$$

3.3.3 Definition 6. Account reachability $AR(s_1, s_2)$ is calculated as:

$$AR(s_1 \rightarrow s_2) = max_{q \subseteq Q}(AR(s_1, s_2, q))$$

$$Q = GenQueries(s_1 \cdot prof, s_1 \cdot msg).$$

In the formula, we define AR from s_1 to s_2 by each query q, and it is calculated by the ratio of $Score(s_1, s_2)$ to the sum of similarity scores between s_1 and the candidates. We define AR as the maximum ratio among all queries in Q.

For example, if we assume that the attacker generates two queries $Q = \{q_1, q_2\}$ from the profiles and messages of s_1. Using query q_1, allows them to select three candidates c_1, c_2, c_3 and the similarity scores with s_1 are 10, 8 and 7, respectively. If c_2 is s_2, the AR by q_1 is $8/(10 + 8 + 7) = 0.32$. Next, they select six candidates c_1 […], c_6 and the similarity scores with s_1 are 30, 2, 2, 1, 1 and 0, respectively. If c_1 is s_2, the AR by q_2 is 0.83. From these results, $AR(s_1, s_2)$ is 0.83 by selecting the highest value of $AR(s_1, s_2, q)$.

If the number of candidates selected by q is small and s_2 is included in $Cand(q)$, $AR(s_1, s_2, q)$ generally becomes large. However, even if the similarity score of $Score(s_1, s_2)$ is high, the value may not be so high when the similarity scores between other candidates and s_1 are also high. On the other hand, even if many candidates are selected by q, $AR(s_1, s_2, q)$ becomes high when similarity score $Score(s_1, s_2)$ is particularly higher than others.

3.4 An implementation of the attack model

In this section, we describe an implementation of the attack model as an example. In the attack model, the attacker should implement three functions: $GenQueries(s_1 . prof, s_1 . msg)$, $Cand(q)$ and $Score(s_1, c)$. To implement them, we first assume the following attack scenario.

3.4.1 Example attack scenario. The attacker uses a search engine to select the candidates of s_2. They first determine a set of keywords from the profiles of s_1. Next, they select a subset of keywords, which they use in the search engine to collect the candidates of s_2. They regard that the higher the account is ranked in the search result page, the more probable it is s_2.

The attack scenario is simple and intuitive, and even an attacker who has no technical skill for inferring can find candidates of s_2. Therefore, we consider that this scenario is suited for the first check of AR.

Based on the scenario, we define the following functions:

- *GenQueries*(s_1 . *prof,* s_1 . *msg*): Let *KeywordSearch*(*ks, engine*) be a function that searches for accounts by using search engine *engine* and a set of keywords *ks*. *GenQueries* outputs a set of functions *Keywords Search* according to sets of keywords *ks*. To build a list of keywords, it first extracts details including s_1's account name, hometown, company name, academic records and favorite things. Any subsets of the keywords are used as *ks* of the function *KeywordSearch*.
- *Cand*(*q*): It obtains the candidates of s_2 by using query q that is *Keyword Search*(*ks, engine*). In most search engines, it is possible to specify the domain of target Web pages. Therefore, it specifies the domain URL of the SNS as a search option and outputs a set of accounts that are listed in the search engine result page (SERP).
- *Score*(s_1, c): We define the score by using the rank in the SERP of the set of keywords *ks* as:

$$Score\,(s_1, c) = \frac{1}{Rank(ks, c)}$$

where *Rank*(*ks, a*) = {Ranking of c in the SERP of *ks*.}

In this example, we assume that the user has two accounts, one is a Facebook account, s_1, and the other is a Twitter account, s_2. The attacker knows the Facebook account, s_1, from which he obtains a set of keywords {*ayano, japan, ochanomizuuniversity*}. From the keywords, the attacker generates subsets of keywords ({*ayano*}, {*japan*}, {*ochanomizuuniversity*}, {*ayano, japan*} […], {*ayano, japan, ochanomizuuniversity*}). For each set of keywords, the attacker searches for Twitter accounts using a search engine. When the attacker searches for accounts based on the set of keywords {*ayano, japan*} using Google, s_2 is ranked first among five accounts in the SERP, *Score*(s_1, s_2) is 1, and the AR by *Keyword Search* ("*Google*", {*ayano, japan*}) is calculated as:

$$AR(s_1, s_2, Keyword\ Search\ (\text{"}Google\text{"},\ \{ayano, japan\})) = \frac{1}{1 + 1/2 + 1/3 + 1/4 + 1/5} = 0.44$$

3.5 Visualization of the result

ARChecker shows the value of AR to inform the user the risk of exposing his personal information. However, it is difficult to understand the risk only from the value of the AR. For example, in the example described in Section 3.4, the AR is 0.44. The user cannot understand whether the value shows a dangerous status or not. In addition, the user

cannot understand how the value is calculated. If the user's status is dangerous, the user should modify his profiles and messages so that they can reduce the risk. ARChecker gives advice on how to decrease the value of the AR.

Therefore, ARChecker visualizes the value of AR using human icons, as shown in Figure 2. To use ARChecker, a user first logs onto two SNS sites (Figure 2①), and presses the check button (Figure 2②). After the system calculates the AR, the value is displayed (Figure 2③). In addition, in the bottom left area of Figure 2④, the value is visualized by human icons, and the right area of Figure 2⑤ shows the list of keywords which are used for searching the candidate's accounts.

All candidate users that are selected in Step 1 of the attack are shown by human icons. The size of each icon depends on score $Score(s_1, c)$, and the icons are arranged in descending order. Notice, that the user's account is depicted by a colored icon, while all other accounts are shown as gray-scaled icons. The value of AR is high when only a few candidates are found or when $Score(s_1, s_2)$ is much bigger than the other icons. The visualization results show such situations. In Figure 2④, the biggest icon is s_2; all others are smaller. The value of AR in this case is 38.8 per cent. A user may first think the value is not so high. However, the visualization result in Figure 2④ may lead the user to realize that the value is higher than first thought. In this way, the user can intuitively make sense of the value from the visualization result.

In the bottom right area of Figure 2, the keywords are displayed in the form of a "tag cloud". The size of a set of words (ks) depends on the value of $AR(s_1, s_2, KeywordSearch(engine, ks))$. In this example, the word "Ayano" is much bigger than the other words. As the large-sized words are the main factors affecting the value of AR, the value may decrease when the user removes those words from his profiles or messages. After modifying the profiles or messages, the user can recheck the value of the AR to see how much the value has decreased. Figure 4 shows the interaction flow for reducing the risk of personal information leakage using ARChecker. We suggest using ARChecker repeatedly until the value becomes sufficiently small.

4. Discussion

In this paper, we introduce a low-level implementation of an attack model. Our experiments show that AR is generally higher than the user expects, even if we use the low-level implementation of the attack model. Nevertheless, attackers may implement more sophisticated methods. In this section, we discuss techniques to develop the similarity measures of two accounts in Step 2 of the attack model.

The implementation described in Section 3.4 relies solely on the scores ranked in the search engine results page, as $Score(s_1, c)$; it does not use the messages and profiles of the account to calculate the similarity between account s_1 and each candidate c.

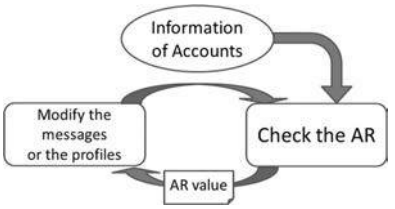

Figure 4.
Interaction flow of ARChecker

If the candidate accounts have user attributes such as gender, age, career and interests, it is possible to calculate the similarity using the user's attributes. However, some SNSs such as Twitter and Flickr provide a free text style field for the user profile, making it difficult to extract the user's attributes. User attribute estimation techniques (Mislove *et al.*, 2010; Rao *et al.*, 2010) estimate the user's attributes from free text-styled profiles, messages and user–user friendships.

Through calculation of the similarity of accounts using authorship identification techniques (Stamatatos, 2009), we can further detect the features of writing style using text mining techniques.

In addition, some SNSs such as Twitter and Flickr have a "geotag" function, which allows users to automatically enter geophysical information with a message or photos. Moreover, users often write messages that include their current location. From such geophysical information, some researchers or attackers can estimate a user's home location (Li *et al.*, 2012).

Friendship relations can also be used for calculating account similarity. Narayanan and Shmatikov (2009) proposed a technique for estimating "the same user" link between two nodes in different social graphs. It uses a link prediction kit to learn the patterns of already-known, "same user" links to find the hidden links.

By implementing these advanced techniques into the attack model, ARChecker can simulate various attacks to protect users from more sophisticated cyber attacks.

5. Model of using local information

In this section, we consider the AR that focuses on local information to be included in the contents of the SNS. Local information is information that is easy to identify an individual; also, it is easily added to the contents of the SNS. Many of the users who use Facebook have registered the residence in their profile. Some users appended local information by using a tag. Twitter users can easily post to where you are now because Twitter is an SNS that can post easily. We focused on local information from these circumstances.

5.1 Definition of formula

We define the following functions:

- *GenQueries*(s_1 . *prof,* s_1 . *msg*): Let *KeywordSearch*(*ks, engine*) be a function that searches for accounts by using search engine *engine* and a set of keywords *ks*. *GenQueries* outputs a set of functions *Keywords Search* according to sets of keywords *ks*. To build a list of keywords, it first extracts the area name from the contents of the post s_1. Any subsets of the keywords are used as *ks* of the function *KeywordSearch*.
- *Score*(s_1, *c*): Let the local information obtained from s_1 be *locate*(s_1), and let the same information as region *locate*(s_1) of the local information obtained from the candidates c_i be *locate*(c_i). Similarity between s_1 and each candidate is calculated by the following formula:

$$Sim(s_1, c_i) = \frac{|locate\,(c_i)|}{|locate\,(s_1)|}.$$

It calculates the temporary *Score* for each candidate based on the above formula. Let it be $PreScore(s_1, c_i)$ as follows:

$$PreScore(s_1, c_i) = \frac{Sim(s_1, c_i)}{\Sigma_{c \in C\,and\,(q)}\, Sim(s_1, c)}.$$

Next, it extracts the candidate account with the *PreScore* value of 0.1 or more, and its re-ranking order of *PreScore* value is large. On the basis of this ranking, let $Score(s_1, c)$ be the value of the inverse of the ranking.

5.2 Calculation example

It is assumed to Twitter and Facebook, the SNS of interest. I will show examples of calculations of *Score* and *GenQueries*.

5.2.1 Example 1 Twitter → Facebook. We assumed that the known account is Twitter, and that the subject to find is Facebook:

- *GenQueries*: It extracted local information from the *tweet*. Let keywords be local name that frequently appears in the *tweet*. It searches the Facebook account by using the search function of Facebook, or by using a search engine like *Google*.
- *Score*: It extracted local name from the Facebook accounts that were obtained as a result of the search. To determine the *Score* using local name obtained earlier and local name extracted from Twitter.

5.2.2 Example 2 Facebook → Twitter. We assumed that the known account is Facebook, and that the subject to find is Twitter:

- *GenQueries*: It extracted local information from the contents of the Facebook account. Let keywords be local name that frequently appears in the contents of the Facebook account. It searches the Twitter account by using a search engine like *Google*.
- *Score*: It extracted local name from the Twitter accounts that were obtained as a result of the search. To determine the *Score* using local name obtained earlier and local name extracted from Facebook.

We verified below about AR calculation model using the profile information, and using local information.

6. Experiments

In this section, we examine the AR calculation model using geographical information and the profile. In this paper, we considered the four methods to calculate AR (Table I).

We examined each method. We assumed that the SNSs that are subject to validation are Twitter and Facebook. For this experiment, we collected 50 pairs of Twitter and

Table I. Methods to calculate account Reachability

	$Score(s_1, c)$	
$GenQueries(s_1\,.\,prof,\ s_1\,.\,msg)$	Section 3	Section 4
Section 3	I	II
Section 4	III	IV

Facebook accounts that are used by the same user. We first categorize the users into the following two categories:

(1) *NotCare*: The users of the category answer that they do not care if a stranger identifies their Twitter account from their Facebook account. There are 26 users in this category.

(2) *NotWant*: The users of this category answer that they do not want their Twitter account to be identified from their Facebook account. There are 24 users in this category.

6.1 Experiments 1: Method I

In this section, we calculate the values of AR from a Facebook account to the corresponding Twitter account using Method I.

We implement the attack model described in Section 3.4. In the *GenQueries*(s_1 . *prof*, s_1 . *msg*) function, we extract the following user attributes from the Facebook account: name, affiliation, birthday and location. In the *Cand*(*q*) function, we use Google to search for the corresponding Twitter account.

Figure 5 shows the values of AR. We list the user's numbers in the horizontal axes. The users from #1 to #26 are *NotCare* category user, and the remainder are *NotWant* category users.

The result shows that the ARs of the *NotCare* category users are between 0 and 40 per cent. The level of implementation applied in this experiment generally scores low values. For example, a Twitter account of User #15 is ranked first among five users in the SERP; therefore, the value is 44 per cent. Although the value of User #7 is only 17 per cent, his Twitter account is ranked second among ten users. We regard that if the value is more than 10 per cent, the user should care. In addition, we should note that three *NotWant* users score more than 10 per cent. User #29 scores 100 per cent because only his account is listed up in the SERP by his affiliation as keywords. Although the other 20 users score 0 per cent, it does not mean they are safe because the level of implementation

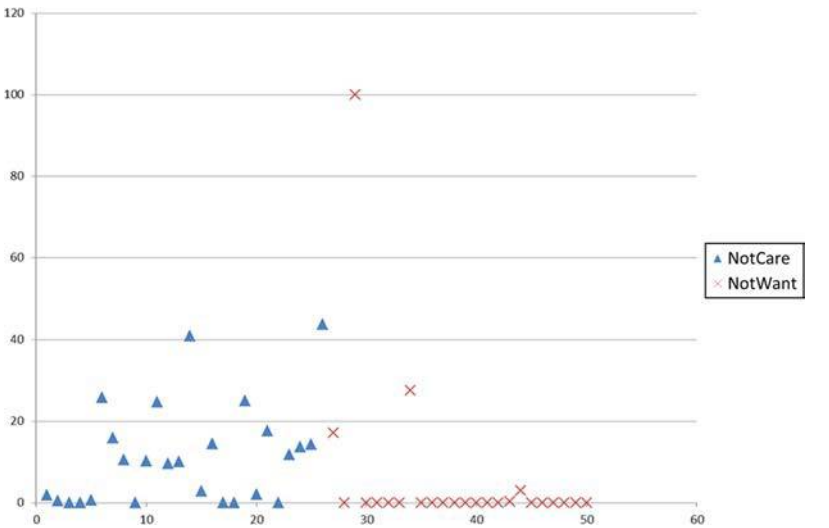

Figure 5.
Values of account reachability using Method I

in the experiment is quite naive. The attacker may use more advanced techniques. We should define more level of implementation to make ARChecker more effective. Most of the users with an AR of 0 per cent write few attributes except their names on their Facebook profiles, which are pseudonyms on their Twitter accounts. Therefore, ARChecker cannot locate the corresponding Twitter account.

Figure 6 shows keyword sets that get the maximum values among any possible keywords. The horizontal axis shows the keywords sets. For example, a user gets 100 per cent by using their affiliations as a keyword set. There are only ten types of keywords sets in Figure 6. Sets of three keywords are deemed unnecessary, as they do not produce a maximum value for any of the accounts. Sixteen users produce the maximum value when they use their name as a keyword because they write their name on their Twitter accounts, in their profiles or in their messages. In addition, seven users produce the maximum value through a combination of their name and another attribute.

6.2 Experiments 2: Method II

In this section, we calculate the values of AR from a Facebook account to the corresponding Twitter account using Method II.

In the *GenQueries*(s_1 . *prof*, s_1 . *msg*) function, we extract the following user attributes from the Facebook account: name, affiliation, birthday and location. In the *Cand*(q) function, we use Google to search for the corresponding Twitter account. In the *Score*(s_1, c) function, we use the formula in Section 4.

Figure 7 shows the values of AR.

The nine users have value of AR. Figure 8 shows the result of comparing Method I and Method II for the nine users.

The eight users have value of AR higher than value of AR using Method I. It is thought that the account that a subject is nearer to was extracted because it looks at the contents of

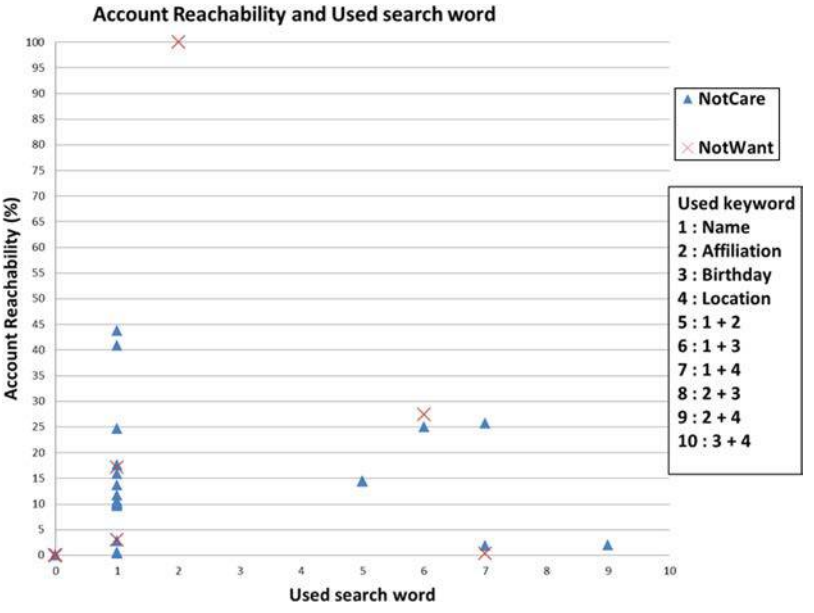

Figure 6.
Keyword sets which achieve the maximum *AR*

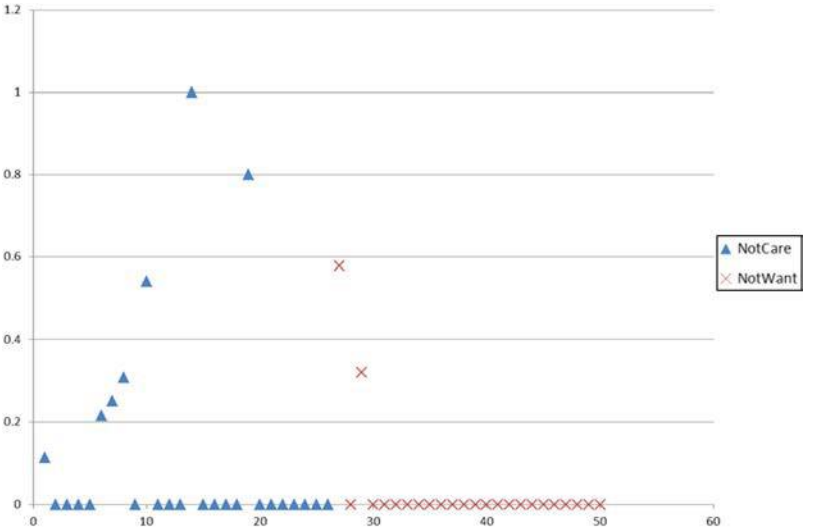

Figure 7.
Values of account reachability using Method II

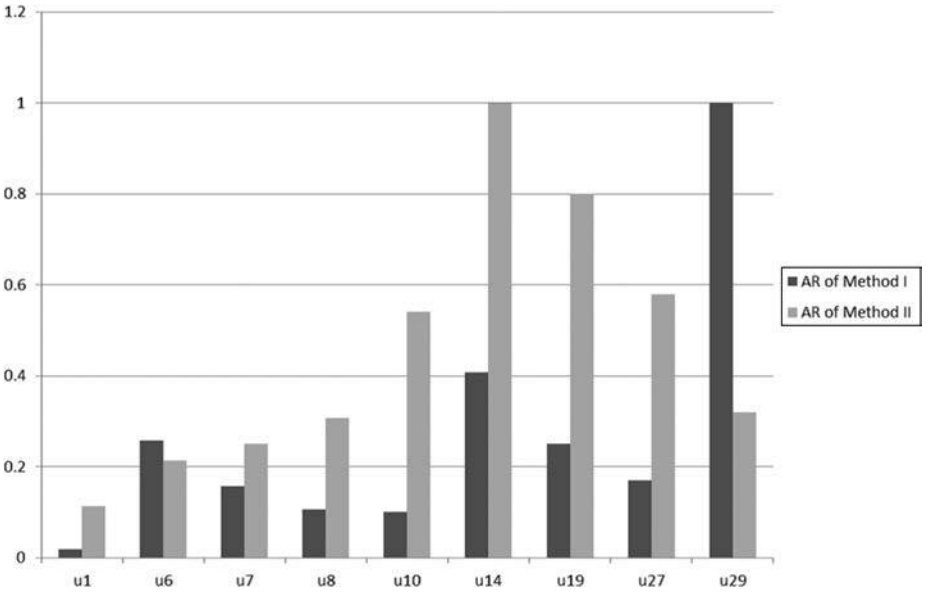

Figure 8.
Comparing values of account reachability using Method I and Method II

tweets of each candidate account and measures a similar degree. User #29 has lower value of AR than Method I because geographical information becomes noise in his AR.

The number of the provided candidate accounts was squeezed in each subject by 1-11 accounts. Figures 9 and 10 show the value of *Score* whose representative account. Figure 9 shows value of *Score* first calculation, and Figure 10 shows value of *Score* second calculation.

Cand	Score
c1	0.25
c2	0
c3	0
c4	0.0625
c5	0
c6	0.25
c7	0
c8	0.0625
c9	0.0625
c10	0.3125

Figure 9.
Value of *Score* first calculation

Cand	Score
c1	0.308
c6	0.308
c10	0.385

Figure 10.
Value of *Score* second calculation

Originally the number of the provided candidate accounts was ten accounts, but this was narrowed down to three accounts after the first *Score* calculation.

6.3 Experiment 3: Method III and IV

First, when it assumed the geographical information that it extracted from the tweet are the keywords, it examined whether it could find the Facebook account of the subject.

Keywords are top five geographical names extracted from the tweets, and a query was generated by combining the keywords. We used the *Google* search engine and friend search feature of Facebook to search.

All subjects did not find information in the *Google* search engine and used the friend search feature of Facebook to search. We cannot search the objective user without searching the custom page because the friend search feature of Facebook goes for the spot search at the same time when we search it with one local name. In addition, it judges one local name with a person's name when it searches in combination more than two local names. It is thought that the user whom we do not know at all in Facebook cannot be found easily. When we use the *Google* search engine, also the personal page cannot be searched easily. Facebook has a lot of personal information. Therefore, it is thought that it limits a search in the Facebook side to prevent personal information from leaking out by an easy search.

Next, when it assumed that the geographical information it extracted from contents of Facebook are the keywords, it examined whether it could find the Twitter account of the subject.

Keywords are top 5 geographical names extracted from contents of Facebook, and a query was generated by combining the keywords. We used the *Google* search engine to search.

All subjects did not find information in the *Google* search engine Top 200 result page. When it searches Twitter account using only geographical information as keywords, there are many candidates because Twitter user tweets a lot of geographical information, for example tweet spot name while traveling.

From these results, we understand it was difficult to find out a different account only using the geographical information.

6.4 User test

We executed user tests to evaluate the usability of ARChecker. ARChecker implemented Method I. The subjects are six people (four women in their 20s, one woman in her 30s, one man in his 20s) who have both Facebook and Twitter accounts. We asked them to use ARChecker and answer the following questionnaire:

Q1. What was your received value of AR?

Q2. How did the result compare with your expectations?

Q3. Did the visualization result of the AR value help to understand your privacy risk?

Q4. Did the word cloud give you advice to modify your profiles and messages to decrease your privacy risk?

Q5. Was the system easy to use?

The answers to the questions are shown in Figure 7 (Figure 11).

Three of the users' AR was 0 per cent because they do not release their user attributes in their Facebook account profile or they restrict access to only their friends. However, users with an AR of 100 per cent post many items of individual data in both accounts. In Question 3, all subjects answer affirmatively. As we described in Section 4.2, the current ARChecker scores low values; therefore, it is difficult to determine whether the user's risk is high. One subject whose score is a low 20 per cent comments that she felt the risk is higher than she first surmised on seeing the visualization result. One subject gives a neutral response to Question 4, commenting that it is difficult to understand where the displayed words are written in her Facebook and Twitter accounts.

7. Conclusions

Following incidents in which SNS users suffered at the hands of cyberstalkers, we considered a way to protect users from cyberstalker attacks. Cyberstalkers can access

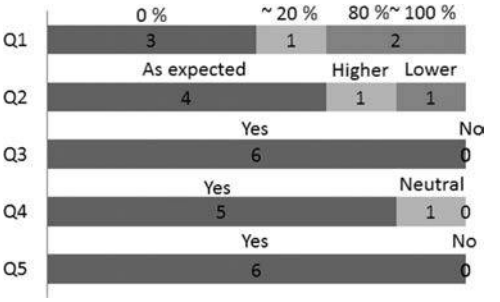

Figure 11.
Results of the user questionnaire

more information through aggregate attributes on a user's multiple accounts. Therefore, our *ARChecker* has been designed to prevent personal information from being leaked due to plurality of associated SNS accounts, and to ensure that it correctly recognizes the information published on users' accounts. The system very simply checks AR and shows the result and reasons for accessibility to personal information. From the results, users can learn how to protect themselves from privacy risk by taking certain measures. Furthermore, it is possible to check whether the measures have been put into proper effect by repeated use. Improvement of the users' understanding over time will continue their use of SNSs with suitable countermeasures.

A number of factors are used to determine the AR. Further developments of the ARChecker will determine more sophisticated methods to evaluate AR, such as information revealed in written posts. In addition, we will discuss the interface and implementation to protect exposure of personal information effectively.

References

Akcora, C.G., Carminati, B. and Ferrari, E. (2012), "Privacy in social networks: how risky is your social graph?", *2012 IEEE 28th International Conference on Data Engineering (ICDE)*, *Washington, DC*, pp. 9-19.

Boyd, D. and Hargittai, E. (2010), "Facebook privacy settings: who cares?", *First Monday*, Vol. 15 No. 8.

Carminati, B., Ferrari, E., Morasca, S. and Taibi, D. (2011), "A probability-based approach to modeling the risk of unauthorized propagation of information in on-line social networks", *Proceedings of the First ACM Conference on Data and Application Security and Privacy*, *San Antonio, TX*, pp. 51-62.

Dubrawski, A., Sarkar, P. and Chen, L. (2009), "Trade-offs between agility and reliability of predictions in dynamic social networks used to model risk of microbial contamination of food", *Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Mining*, *Athens*, pp. 125-130.

Forbes (2011), "Multiple personalities and social media: the many faces of me", available at: www.forbes.com/sites/meghancasserly/2011/01/26/multiple-personalities-and-social-media- the-many-faces-of-me/

Irani, D., Webb, S., Pu, C. and Li, K. (2011), "Modeling unintended personal-information leakage from multiple online social networks", *IEEE Internet Computing*, Vol. 15 No. 3, pp. 13-19.

Joinson, A.N. (2008), "Looking at, looking up or keeping up with people?: motives and use of facebook", *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, *New York, NY*, pp. 1027-1036.

Li, R., Wang, S., Deng, H., Wang, R. and Chang, K.C.-C. (2012), "Towards social user profiling: unified and discriminative influence model for inferring home locations", *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1023-1031.

Machanavajjhala, A., Kifer, D., Gehrke, J. and Venkitasubramaniam, M. (2007), "L-diversity: privacy beyond k-anonymity", *ACM Transactions on Knowledge Discovery from Data*, Vol. 1 No. 1.

Mislove, A., Viswanath, B., Gummadi, K.P. and Druschel, P. (2010), "You are who you know: inferring user profiles in online social networks", *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, *New York, NY*, pp. 251-260.

Narayanan, A. and Shmatikov, V. (2009), "De-anonymizing social networks", *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, Berkeley, CA, pp. 173-187.

Pulse Compete Pulse (2007), "Connecting the social graph: member overlap at OpenSocial and facebook", available at: https://blog.compete.com/2007/11/12/connecting-the-social-graph-member-overlap -at-opensocial-and-facebook/

Rao, D., Yarowsky, D., Shreevats, A. and Gupta, M. (2010), "Classifying latent user attributes in twitter", *Proceedings of the 2nd International Workshop on Search and Mining User-Generated Contents*, *Toronto*, pp. 37-44.

Sadilek, A., Kautz, H. and Bigham, J.P. (2012), "Finding your friends and following them to where you are", *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, *Rochester, NY*, pp. 723-732.

Stamatatos, E. (2009), "A survey of modern authorship attribution methods", *Journal of the American Society Information Science and Technology*, Vol. 60 No. 3, pp. 538-556.

Sweeney, L. (2002), "k-anonymity: a model for protecting privacy", *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, Vol. 10 No. 5, pp. 557-570.

Wang, T., Srivatsa, M., Agrawal, D. and Liu, L. (2011), "Modeling data flow in socio-information networks: a risk estimation approach", *Proceedings of the 16th ACM Symposium on Access Control Models and Technologies*, *Innsbruck*, pp. 113-122.

Yoshikuni, A. and Watanabe, C. (2013), "Account reachability: a measure of privacy risk for exposure of a user's multiple SNS accounts", *Proceedings of International Conference on Information Integration and Web-based Applications & Services*, *New York, NY*, p. 542.

Zheleva, E. and Getoor, L. (2011), "Privacy in social networks: a survey", in Aggarwal, C. (Ed.), *Social Network Data Analytics*, Chapter 9, Springer, pp. 247-276.

Corresponding author
Ayano Yoshikuni can be contacted at: yoshikuni.ayano@is.ocha.ac.jp