Defending against User Identity Linkage Attack across Multiple Online Social Networks

Yilin Shen
Samsung Research America
75 West Plumiera Dr.
San Jose, CA 95134, USA
yilin.shen@samsung.com

Fengjiao Wang*
Dept. of Computer Science
University of Illinois at Chicago
Chicago, IL 60607, USA
fwang27@uic.edu

Hongxia Jin
Samsung Research America
75 West Plumiera Dr.
San Jose, CA 95134, USA
hongxia.jin@sisa.samsung.com

ABSTRACT

We study the first countermeasure against user identity linkage attack across multiple online social networks (OSNs). Our goal is to keep as much as user's information in public and meanwhile prevent their identities from being linked on different OSNs via k-anonymity. We develop a novel greedy algorithm, incorporating an efficient manner to compute the greedy function, and validate it in terms of both solution quality and robustness using real-world datasets.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

Keywords

User Identity Linkage Attack, Countermeasure, Algorithm

1. INTRODUCTION

More and more Online Social Networks (OSNs) have emerged for various purposes. People provide specific profiles, interact with particular types of friends and join distinct groups on different OSNs. However, a person's multiple user identities can be easily linked by a practical *User Identity Linkage Attack*, leading to user privacy breach and put the users at risks unconsciously [2]. For instance, when the attacker links a user's two identities on Facebook and Linkedin, his/her professional connections on Linkedin are disclosed to the attacker, although they are not his/her friends on Facebook. More seriously, based on the integrated profile and inferred private information from the friends of the linked user, the attacker could create fake accounts pretending to be this person on social networks and then solicit others to connect, referred to as an alternative identity theft. Therefore, it is of great importance to prevent the user identity linkage across OSNs by malicious users beforehand.

In this paper, we develop the first countermeasure framework against user identity linkage attack [1] across multiple OSNs, in which we formulate an optimization problem to study the trade-off between user information sharing and user privacy via k-anonymity. We devise an effective algorithm to solve this problem and validate its performance and robustness on real-world datasets.

2. PROPOSED COUNTERMEASURE

We propose a countermeasure aiming to defend against the user identity linkage (UIL) attack, which can smartly generate features merely based on user's public information, thereby only needs 0.5% of user linkage information to accurately predict over 85% other user identity linkages. Our countermeasure is applied to each user on OSN1 (service OSN) by smartly hide his information for preventing him from being linked to their identities on OSN2 (auxiliary OSN).

Problem Definition: In order to tackle the trade-off between user information sharing and user privacy, we define an optimization problem, named *PRivacy-aware User iNformation sElection* (PRUNE) problem as follows: *Given a positive integer k, a user $\mathfrak{w}$ associated with two identities $\mathfrak{w}_1$ and $\mathfrak{w}_2$ on OSN1 and OSN2 respectively, i.e., $\mathfrak{w} = (\mathfrak{w}_1, \mathfrak{w}_2)$. The objective of PRUNE problem is to maximize user $\mathfrak{w}$'s information visibility in OSN1(service OSN) and achieve the k-anonymity for user $\mathfrak{w}$ between his identity $\mathfrak{w}_1$ and all user identities V_2 on OSN2 (auxiliary OSN).*

More specifically, a user $\mathfrak{w}$ achieves k-anonymity if and only if there exist at least $k-1$ users S (identical users) on OSN2 such that $m(\mathfrak{w}_1, \mathfrak{w}_2) \leq m(\mathfrak{w}_1, u)$ for any $u \in S$, where $m(\cdot,\cdot) = \sum_{i \in \mathfrak{F}} \beta_i(\cdot,\cdot)$ is determined by Adaboost (Decision Stump) classifier on features due to its most serious consequence in UIL attack. Here $\beta_i(\cdot,\cdot)$ is the summation of coefficients in Adaboost classifier for each feature. The maximization of user information visibility is to $\max \sum_{i \in S} v_i^{\mathfrak{w}}$, in which $v_i^{\mathfrak{w}} = 1$ means that user $\mathfrak{w}$ prefers to release information i to public and 0 otherwise.

Resist User LinkagE (RULE) Algorithm: We design the RULE algorithm based on our observation that most of users on OSN2 are *distinctive users*, i.e., they have $\beta_i(\mathfrak{w}_1, \mathfrak{w}_2) \geq \beta_i(\mathfrak{w}_1, u)$ for all features between themselves and user $\mathfrak{w}$. Therefore, our proposed RULE algorithm iteratively hides a feature i which maximizes

$$\lambda(i) = \sum_{u \in V_2^j} \mathcal{I}(\beta_i(\mathfrak{w}_1, \mathfrak{w}_2) - \beta_i(\mathfrak{w}_1, u))/v_i^{\mathfrak{w}}$$

where V_2^j is the set of non-identical users on OSN2 after iteration j. The indicator variable $\mathcal{I}(x) = 1$ if $x > 0$ and 0 if $x = 0$. In order to efficiently find the feature i with maximum $\lambda(i)$ in each iteration, we propose three priority queues with respect to the distinctive and non-distinctive users on OSN2 and feature sets: the priority of Q_d is $\sum_{i \in \mathfrak{F}} \mathcal{I}(\beta_i(\mathfrak{w}_1, \mathfrak{w}_2) - \beta_i(\mathfrak{w}_1, u))$; the priority of Q_n is $\sum_{i \in \mathfrak{F}} \beta_i(\mathfrak{w}_1, \mathfrak{w}_2) - \beta_i(\mathfrak{w}_1, u)$; the priority of Q_f is $\lambda(i)$.

*Fengjiao Wang contributed to this work during her internship with Samsung Research America.

Copyright is held by the author/owner(s).

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577208.

In each iteration, RULE algorithm (Algorithm 1) picks the feature in Q_f with highest priority and updates Q_d and Q_n after removing the selected feature. The algorithm terminates if the number of users in Q_d with priority 0 plus the number of users in Q_d with non-positive priority is larger than or equal to k. Otherwise, if there are users in Q_n converting to distinctive users, remove them from Q_n and insert to Q_d. In Q_f, we recompute the feature with highest priority if some users in Q_d becomes to identical users to $\mathfrak{w}$, i.e., their priorities change to 0. If its new priority is invalid, we remove it from Q_f and insert into the existing order. The process terminates until there is a valid feature in Q_f. Note that in most cases, we only need to compute once to update Q_f and this increases the efficiency of our RULE algorithm.

Algorithm 1: RULE Algorithm

```
Input  : $\beta_i(\mathfrak{w}_1, \mathfrak{w}_2), \beta_i(\mathfrak{w}_1, \cdot)$, $v^{\mathfrak{w}}$
Output: visible features $S$
$S \leftarrow \mathfrak{F}$;
Construct priority queues $Q_d$, $Q_n$, $Q_f$;
$D \leftarrow \{u|\text{priority}(u) = 0, u \in Q_d\}$;
Remove $D$ from $Q_d$;
$N \leftarrow \{u|\text{priority}(u) \leq 0, u \in Q_n\}$;
while $|D \cup N| < k$ do
    Hide feature $i$ with highest priority in $Q_f$ ($S \leftarrow S \setminus \{i\}$);
    Update $Q_d$ and $Q_n$ after removing feature $i$;
    Insert $\{u|u \in Q_n, \beta_i(\mathfrak{w}_1, \mathfrak{w}_2) > \beta_i(\mathfrak{w}_1, \cdot) \, \forall i \in \mathfrak{F} \setminus S\}$ into
    $Q_d$;
    $N \leftarrow \{u|\text{priority}(u) \leq 0, u \in Q_n\}$;
    $D \leftarrow D \cup \{u|\text{priority}(u) = 0, u \in Q_d\}$;
    Remove $D$ from $Q_d$;
    Update $\min_{i \in \mathfrak{F} \setminus S} \lambda(i)$ in $Q_f$ until validation;
return $S$;
```

The running time of RULE algorithm is $O(|\mathfrak{F}|(|\mathfrak{F}|\log|\mathfrak{F}| + |V_2|\log|V_2|))$ in the worst case. Note that a more careful implementation can reduce the running time to $O(|\mathfrak{F}|(|\mathfrak{F}| + |V_2|))$ by maintaining Q_d, Q_n and Q_f within time $O(|\mathfrak{F}|)$ and $O(|V_2|)$ respectively. In addition, RULE algorithm achieves k-anonymity of user $\mathfrak{w}$ for any $k > 0$.

3. EXPERIMENTAL RESULTS

Datasets: We collect data (Table 1) from three of the most popular OSNs, Google+, Twitter and Foursquare. Due to the infeasibility of collecting the information of all users in these OSNs, we consider the user identity linkage between *ego users*, i.e., the users we have crawled their complete profile and neighborhood information. Then we link the ego users on these three OSNs using their public API functions. Finally, we link all neighborhood of ego users, including friends, followers and followees, between these three OSNs.

Table 1: Dataset Statistics

User Types		**Ego Users**	**Neighborhood**	
			Followers	Followees
OSNs	Google+	258	86,652	86,297
	Twitter	48,100	11,485,756	6,319,928
	Foursquare	5,709	138,872	138,872
Linked	⟨Twitter,G+⟩	258	875,054	1,459,621
Crossing	⟨Fsq,Twitter⟩	5,709	11,448,961	6,352,896
Users	⟨G+,Fsq⟩	153	71,664	67,797

Performance & Robustness: For each pair of OSNs (OSN1 and OSN2), we consider each one of them as service OSN (and the other as auxiliary OSN) respectively, that is, we run our algorithm on OSN1 and OSN2 separately. Therefore, we will evaluate the proposed algorithm on six datasets. In terms of the inputs, matching function is derived from UIL attack with Adaboost (Decision Stump) as we discussed in the above section. As for the visibility parameters, *w.l.o.g.*, we randomly select either 0 or 1 for each of the user's information.

We compare the solution quality of RULE algorithm with the optimal solution, provided via Integer Programming:

$$\begin{array}{ll} \max & \sum_{i \in \mathfrak{F}} v_i^{\mathfrak{w}} x_i \\ \text{s.t.} & \sum_{i \in \mathfrak{F}} [\beta_i(\mathfrak{w}_1, \mathfrak{w}_2) - \beta_i(\mathfrak{w}_1, u)] x_i \leq \Omega(1 - y_u) \quad \forall u \in V_2 \\ & \sum_{u \in V_2} y_u \geq k \\ & x_i, y_u \in \{0, 1\}, \forall i \in \mathfrak{F}, u \in V_2 \end{array}$$

where $x_i = 1$ if feature i of user $\mathfrak{w}$ is visible and 0 otherwise; $y_u = 1$ if $m(\mathfrak{w}_1, \mathfrak{w}_2) \leq m(\mathfrak{w}_1, u)$ and 0 otherwise; V_2 is the set of users on OSN2 and Ω is a large enough constant. The objective is to maximize the user information visibility as described in problem definition. The first constraint ensures $m(\mathfrak{w}_1, \mathfrak{w}_2) \leq m(\mathfrak{w}_1, u)$ for each user $u \in S$ in the k-anonymity, i.e., $y_u = 1$ for $u \in S$. The second constraint guarantees the k-anonymity. This IP formulation is implemented using the CPLEX optimization suite from ILOG.

Table 2: Performance of RULE (visibility(%))

Dataset		**RULE**	**Optimality**
Service OSN1	Auxiliary OSN2		
Twitter	Foursquare	**99.231**	99.231
	Google+	99.578	99.582
Google+	Foursquare	**99.202**	99.202
	Twitter	83.287	83.620
Foursquare	Google+	**47.729**	47.729
	Twitter	49.254	49.312

We test our RULE algorithm on these six datasets, with different k values from 90% to 100% users on OSN2. As shown in Table 2, the visibility returned by RULE algorithm is at most 0.5% smaller than optimal solution, and reaches optimality in 3 datasets. And the running time of RULE algorithm is consistently less than 10 seconds, which is up to 20 times faster than solving the IP formulation.

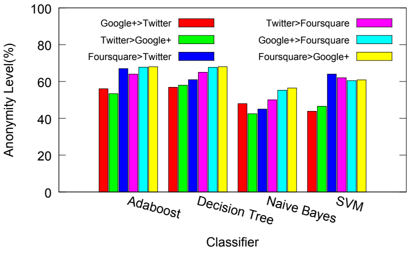

Figure 1: Robustness of RULE

Figure 1 shows the robustness of RULE algorithm by measuring anonymity level with respect to different classifiers (service OSN > auxiliary OSN). As this anonymity level remains consistently larger than 40%, the application of our countermeasure can still anonymize a user with at least other 36% users on average as we select k between 90% and 100% of all users in our experiments.

4. REFERENCES

[1] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of WWW '09*, pages 551–560, 2009.

[2] D. Irani, S. Webb, K. Li, and C. Pu. Large online social footprints–an emerging threat. In *Proceedings of CSE '09*, pages 271–276, Washington, DC, USA, 2009.