A Friendship Privacy Attack on Friends and 2-Distant Neighbors in Social Networks

Lei Jin, Xuelian Long, James Joshi School of Information Sciences, University of Pittsburgh {lej17, xul10, jjoshi}@pitt.edu

ABSTRACT

In an undirected social graph, a friendship link involves two users and the friendship is visible in both the users' friend lists. Such a dual visibility of the friendship may raise privacy threats. This is because both users can separately control the visibility of a friendship link to other users and their privacy policies for the link may not be consistent. Even if one of them conceals the link from a third user, the third user may find such a friendship link from another user's friend list. In addition, as most users allow their friends to see their friend lists in most social network systems, an adversary can exploit the inconsistent policies to launch privacy attacks to identify and infer many of a targeted user's friends. In this paper, we propose, analyze and evaluate such an attack which is called Friendship Identification and Inference (FII) attack. In a FII attack scenario, we assume that an adversary can only see his friend list and the friend lists of his friends who do not hide the friend lists from him. Then, a FII attack contains two attack steps: 1) friend identification and 2) friend inference. In the friend identification step, the adversary tries to identify a target's friends based on his friend list and those of his friends. In the friend inference step, the adversary attempts to infer the target's friends by using the proposed random walk with restart approach. We present experimental results using three real social network datasets and show that FII attacks are generally efficient and effective when adversaries and targets are friends or 2-distant neighbors. We also comprehensively analyze the attack results in order to find what values of parameters and network features could promote FII attacks. Currently, most popular social network systems with an undirected friendship graph, such as Facebook, LinkedIn and Foursquare, are susceptible to FII attacks.

Categories and Subject Descriptors

K.4.1 [Computer and Society]: Public Policy Issues – *privacy*; K.4.2 [Computer and Society]: Social Issues

General Terms

Experimentation, Security

Keywords

Privacy, Social Networks, Random Walk with Restart

1. INTRODUCTION

In online social network systems (SNSs), such as Facebook and LinkedIn, friendship links are usually undirected. A friendship link involves two users and the relationship is visible in both users' friend lists. Both users can separately control its visibility to other users, and hence, a conflict between privacy policies may arise. Such inconsistency of the policies could cause the violation of a user's privacy settings. In a SNS, such as Facebook, a friendship between user A and C is not visible to another user B only when both A and C do not allow B to view it. When only A (C) forbids B to see the link, B can find that A(C) is in C's (A's) friend list if

C(A) allows B to view his friend list. Since the friendship graph is an undirected network, B can get the conclusion that A and C are friends. In this case, A's (C's) privacy policy for B is violated.



Figure 1. An example of the FII attack

Most users in Facebook, LinkedIn and Foursquare allow their friends to see their friend lists since it is the default policy and users tend not to change the default privacy policies [34, 35]. Then, an adversary can exploit the inconsistent policies for the visibility of the friend lists to identify and infer many of a targeted user's friends. For example, as shown in Figure 1, let user *B* be the adversary and user *T* be his target whose privacy settings do not allow *B* to see his friend list. User *A*, *D* and *E* are *B*'s friends and they allow *B* to view their friend lists. *A*, *C* and *D* are actually *T*'s friends but initially *B* is not supposed to know that. However, by checking *A*'s and *D*'s friend lists, *A* can directly identify that *A* and *D* are *T*'s friends. In addition, using the random walk based approach (introduced in Section 3.3.2) on this social graph, *A* can infer that *C* has a high probability of being a friend of *T*. Now, all of *T*'s friends are identified and inferred by *B*.

A user's friend list in a social network is usually valuable and sensitive information for the user [1-4]. For instance, Facebook's move to suspend Google's Friend Connect program's access to Facebook's social graph in 2008 was motivated by the perceived importance of ensuring the privacy of friendship links [1]. In LinkedIn, when a user connects with several well-known professionals, it may help in his search for jobs. At the same time, exposure of a user's friend list may also play against his opportunities, for instance, if it includes an individual with a bad reputation. In addition, if an adversary finds a user's friend list, he may leverage it to launch further, more dangerous privacy attacks. For example, the adversary may infer the user's private attributes in his profile based on the public information of his friends [2, 5, 6]; the adversary can also successfully launch a profile cloning or identity clone attack on a target user whose friend list is exposed[7, 81.

In this paper, we propose a *Friendship Identification and Inference (FII)* attack that is to identify and infer a target's friends in undirected social networks. This attack has two attack steps: 1) friend identification and 2) friend inference. In the friend identification step, an adversary identifies a target's friends based on the adversary's friend list and the friend lists of his friends that are authorized to him to view. The friend inference step involves inferring a target's friends using the random walk with restart approach based on a small segment of the social graph an adversary knows. We demonstrate the effectiveness of the *FII* attack on three different real social network datasets using various measurements. We also comprehensively analyze the attack results based on the social network features, such as the degree and the clustering coefficient of a user. We notice that most popular SNSs with an undirected friendship graph, such as Facebook, LinkedIn and Foursquare, are currently susceptible to *FII* attacks. The contributions of this paper can be summarized as follows:

- We discuss the issues of inconsistency in users' privacy policies for a friendship link between them. Based on such vulnerability, we propose a *FII* attack that aims to identify and infer a target's friends in undirected social networks. Our experimental results show that *FII* attacks are efficient and effective when adversaries and targets are friends or 2-distant neighbors.
- We comprehensively analyze various attack results and try to identify what values of parameters and network features can be leveraged for more successful *FII* attacks. For example, our analysis suggests that an adversary with a higher node degree, a higher clustering coefficient value, more mutual friends between him and the target, etc., has a higher probability to conduct more successful *FII* attacks.

The remainder of the paper is organized as follows. We first present the related work in Section 2 and then propose the *FII* attack and its associated algorithms in Section 3. In Section 4, we present simulations of the attacks using three real social network datasets. We also comprehensively analyze the attack results to identify what values of parameters and social network features can promote the attacks in this section. Finally, we conclude the paper with a discussion of future work in Section 5.

2. RELATED WORK

Many research efforts have focused on the link prediction/inference issues in social networks based on the users' attributes. Hasan and Zaki summarize the supervised link prediction algorithms, including Bayesian probabilistic models, linear algebraic models and probabilistic relational models in [17]. Yin et al. introduce Social-Attribute Network (SAN), an attributeaugmented social network, to integrate network structure and node attributes to perform the link recommendations using the random walk with restart algorithm [12, 18]. Gong et al. extend the SAN framework based on their observation that inferring attributes could help predict links [32]. However, the above approaches usually require users' attribute information for link prediction. In a FII attack, an adversary can launch the attack without knowing users' attribute information.

Recently, some recent efforts have focused on predicting missing links in a global social network. Korolova *et al.* mention that users' behavior of sharing their friends to the public could allow an attacker to discover the topology of a social network [1]. Effendy *et al.* extend the work by Korolova *et al.* and show that such an attack can be "magnified" substantially with the inference on user degrees [20]. Bonneau *et al.* report that the social network structure can be disclosed when a random sample of *k* edges from

each node in the social network is exposed [21]. Leroy *et al.* use group information to build a bootstrap probabilistic graph in order to perform friendship link inferences [22]. Kim and Leskovec adopt the Expectation Maximization (EM) algorithm to infer the missing nodes and edges by observing part of the network [23]. Erdös *el al.* propose a heuristic approach to discover the missing links based on common neighbors of sub-networks [24]. Chen *et al.* construct the Markov Logic Network to infer the friendship links on a large scale social graph [26]. However, the *FII* attack focuses on a local network and the goal of an adversary is to identify and infer a target's friends based on only his friend lists and the friend lists from his friends that are accessible. The *FII* attack does not require as much knowledge of the global social network as the above approaches.

Some existing works propose attacks using the visible local network to identify and/or infer users' private information. Puttaswamy et al. introduce intersection attacks based on shared contents, such as URLs, to infer users' attributes and preferences [25]. The intersection attacks require an adversary to have two compromised nodes in a social graph and then the adversary can infer the value of the resource of a common user (victim) of the compromised nodes. Jin et al. propose the mutual-friend based attack to identify a target's friends via various mutual friend queries among users. Compared to the FII attack, the adversary in above attacks has to know the common features between users. For example, in a mutual-friend based attack, the adversary can query mutual friends between him and any other users. However, mutual friend queries are not necessary for a FII attack. In addition, we believe that a FII attack and a mutual-friend based are complimentary and they can be used together for more successful privacy attacks.

In a *FII* attack, we use the random walk with restart algorithm to infer friends of a target. In the literature, several approaches have been proposed that can be used to infer friends of a target, such as the approaches based on the various similarities among users [16, 17, 29] and the EM algorithm [23]. We do not adopt these approaches because an adversary in a *FII* attack does not need to know any of a target's friends initially. Hence, the similarity between the target and any other node is difficult to compute and the EM algorithm is difficult to apply.

The algorithms proposed in our work can also be used as one of measurements to evaluate the risk of a user's social graph [27] and the privacy risks related to friendships in social networks [28].

3. ATTACK MODEL AND ALGORITHMS

In this section, we formalize the *FII* attack and introduce the attack steps and the corresponding attack algorithms.

3.1 Basic Definitions in Social Networks

A social network is generally modeled as a graph G(V, E), where V represents a set of users and $E = \{(x, y) \mid x, y \in V\}$ represents a set of friendship links among users. In an undirected social graph, such as Facebook and LinkedIn, an edge e = (x, y) is added to E when a friend request from x to y or from y to x is accepted. Such an undirected social graph G can also be represented by an adjacency matrix $A_{n \times n}$ where n = |V|. For each $a_{ij} \in A$, $a_{ij} = 1$ if there is an edge $e = (i, j) \in E$; otherwise $a_{ij} = 0$. Let $P_{n \times n}$ be a transition matrix, where p_{ij} represents the probability of stepping on node j from node i and $p_{ij} = a_{ij} / \sum_i a_{ij}$.

For each user $i \in V$, the user set $F(i) = \{ j \mid j \in V, a_{ij} = 1 \}$ is the friend list of *i*. Note that in an undirected social graph, when $a_{ij} = 1, a_{ji} = 1, j \in F(i)$ and $i \in F(j)$. The user set $D_2(i) = \{ j \mid \exists k, j \in V, a_{ij} = 0, a_{ik} = 1 \text{ and } a_{kj} = 1 \}$ represents the 2-distant neighborhood relationship in an undirected social network is also bidirectional.

In a SNS, such as Facebook, a user usually can see his friend list and friend lists of his friends if he is authorized. However, a user usually cannot see the friend list of another user who is not his friend. In this paper, we assume that a user can only see his friend list and friend lists of his friends who authorize the user to view them. The user cannot see friends of another user who are not friend of the user. Hence, we define an edge set K(i) as the initial knowledge of a user *i*. K(i) includes *i*'s friendship links and the friendship links of *i*'s friends who allow *i* to view their friend lists. For example, in Figure 2, user A has two friends B and C. B allows A to see his friend list while C hides his friend list from A. In this case, $K(A) = \{(A, B), (A, C), (B, D), (B, E)\}$. A does not know the edge (E, F) as A is not E's friend. In addition, A does not know the edge (C, D) since C conceals his friend list from A. Note that A can identify the edge (A, C) = (C, A) from his friend list. Assume there are m distinct nodes in K(i). We also use the adjacency matrix $Ai_{m \times m}$ to represent K(i) and adopt $Pi_{m \times m}$ to represent the transition matrix for Ai.



Figure 2. Initial knowledge of user A

3.2 Definition of the *FII* **Attack**

We present the definition of FII attack as follows:

Definition of the *FII* **Attack**: Given an undirected social network G(V, E), an adversary $b \in V$ and a target $t \in V$,

Privacy Requirement of the Target:

• *t* defines a policy that does not allow *b* to see *F*(*t*);

Adversary's Knowledge before Launching the Attack:

- When b and t are friends, we assume that b only knows his friend list and the friend lists of b's friends except t. K(b) = {(b, x) | x ∈ F(b)} ∪ {(x, y) | x ∈ F(b), x ≠ t, y ∈ V}
- When *b* and *t* are not friends, we assume that *b* only knows his friend list and all friend lists belonging to his friends. $K(b) = \{(b, x) | x \in F(b)\} \cup \{(x, y) | x \in F(b), y \in V\}$

Privacy Attack:

We say that *t* is a victim (compromised¹ node) of a *FII* attack by *b* if *b* obtains $F_b(t) \neq \emptyset$ based on calculations on K(b). Here $F_b(t)$ represents the friends of *t* identified and inferred by *b* through the *FII* attack.

Note that in a *FII* attack, we assume that additional actions, such as the search of public available friend lists and mutual friend based queries, are not available to extend K(b) defined above. As we mentioned in the related work, we notice that *b* may have a more successful attack using some additional actions but such attack scenarios are not considered in this paper.

Next, we present theorems that characterize the key properties of a *FII* attack. Theorem 1 characterizes the necessary precondition for a *FII* attack to be successful and the victim set an adversary can compromise; Theorem 2 characterizes the size of the identified and inferred friends of t by b.

Theorem 1: *b* can compromise *t* in a *FII* attack if and only if *t* is either a friend or a 2-distant neighbor of *b*. Let PV(b) be a set of potential victims that *b* can compromise through *FII* attacks in a social graph; then $PV(b) = F(b) \cup D_2(b)$.

Theorem 2: When *b* can compromise *t*, $F_b(t) \subseteq IS_b(t)$, where $IS_b(t)$ represents the ideal exposed friends in a *FII* attack. $IS_b(t) = \{b\} + \{(F(b) \cup D_2(b)) \cap F(t)\}$ when *b* and *t* are friends. $IS_b(t) = (F(b) \cup D_2(b)) \cap F(t)$ when *b* and *t* are 2-distant neighbors.

For example, as shown in Figure 3, user *B* is the adversary and user *T* is his target. Users *A*, *C*, *D* are the potential identified and/or inferred friends of *T*. As shown, all of them are included in $IS_B(T) = (F(B) \cup D_2(B)) \cap F(T)$.

B is the adversary and T is his target



Figure 3. F(T) and the user set by F(B) and D₂(B)

In a *FII* attack, there are two attack steps: 1) friend identification and 2) friend inference. In the friend identification step, the adversary *b* aims to identify *t*'s friends by checking whether *t* exists in the edges in K(b). In the friend inference step, *b* tries to infer a *t*'s friends using the random walk with restart approach based on the results from the friend identification step and the knowledge *b* has as per the definition above. We describe these two attack steps in detail in Section 3.3.

¹ In this paper, when we say b compromises t in a *FII* attack, it means that b can successfully identify and/or infer t's friends during the attack.

3.3 Attack Steps and Algorithms

3.3.1 Friend Identification

The purpose of the friend identification step is to identify a target's friends based on the initial knowledge that an adversary has. In this step, an adversary *b* is able to check whether a target *t* exists in the edges in K(b). When $(b, t) \in K(b)$, *b* is *t*'s friend. When $\exists x \in F(b)$ and $(x, t) \in K(b)$, *x* is one of *t*'s friends. The algorithm of the friend identification is shown as follows.

Algorithm 1 Friend Identification

Input : an adversary b , a target t , $F(b)$	$(b), D_2(b), K(b)$

Output : the identified friend set $FID_b(t)$, AdjacencyMatrix Ab
1. $F_b(t) = \emptyset$
2. if $t \in F(b)$ or $t \in D_2(b)$
3. $Ab = \text{convertToAdjacencyMatrix}(K(b))$
4. $m = \operatorname{ncol}(Ab) // \operatorname{number} of distinct nodes in Ab$
5. // find the row which represents the link relations between t
6. // and other nodes in Ab
7. $i = indexOf(Ab, t)$
8. for $j=1$ to m do
9. if $a_{ij}=1$ then
10. // get the vertex from the index of the matrix
11. $f = getNodeFromIndex(Ab, j)$
12. add f to $FID_b(t)$.
13. end if
14. end for
15. end if

Note that all identified friends are actually friends of t in this step.

3.3.2 *Friend Inference*

In the friend inference step, an adversary b adopts a random walk based approach to infer friends of a target t based on the adjacency matrix Ab composed by K(b).

The random walk based approaches are proved to be effective for predicting friend and users' trajectory in social networks [12-15]. In this paper, we adopt the approach of random walk with restart [30] to infer a target's friends. With regards to the friend inference step, we note that, intuitively, if a node is closer to the target, has a higher degree and has more mutual friends with a target, then it is more likely that the node is a friend of the target. All these factors are counted in random walk with restart [13, 30]. In the random walk with restart approach, a node follows the random walk with the probability of *a* but it jumps back to the originator with probability 1-a. After doing the random walk for a long period of time, the transaction matrix of such a graph, which represents the probabilities of links between each pair of nodes, will be stationary distribution. An adversary can then infer the target's friends based on such the stable transaction matrix.

In particular, an adversary b has to complete the following steps to infer a target t's friends:

- 1. Get the adjacency matrix Ab and $FID_b(t)$, which represents the identified friends, from the friend identification step. Compute the transition matrix Pb based on Ab.
- Define the initial vector V₀ =(v_i)_m, where m is the number of distinct nodes in Ab. For each v_i, when i is the index of t in Ab, v_i = 1; otherwise, v_i = 0.

- 3. Set a vector $R = (r_i)_m$ with all "0" for each element except the j^{th} element (the index of *t* in *Ab*) for which we set "1".
- 4. Define a parameter $a \in (0, 1)$. Define V_n as a distribution for t after n times walking. $V_n = a \times Pb \times V_{n-1} + (1 a) \times R$. Set second parameter ε which represents a very small value, *e.g.*, 10^{-6} . Keep walking till $|V_n V_{n-1}| < \varepsilon$. In this case, we say V_n is a stationary distribution and it reflects the probabilities of the link between t and any other nodes in Ab.
- 5. Set the third parameter $\beta \in (0, 1)$, which is used to determine the total number of users inferred as *t*'s friends. The number of inferred friends for *t* is $[\beta \times m]$. Choose $[\beta \times m]$ distinct nodes from *Ab* based on the result of V_n . These nodes have the higher probabilities for *t* than for the remaining nodes. However, the selected nodes should not be already included in *FID_b(t)* and none of them should not be equal to *t*. These chosen nodes are inferred friends of *t* and we add them into *FIN_b(t)*.
- 6. Finally, $F_b(t) = FID_b(t) \cup FIN_b(t)$.

The algorithm used for the friend inference is shown below.

Algorithm 2 Friend Inference

Input: an adversary *b*, a target *t*, *Ab*, $FID_b(t)$ from Algorithm 1, parameters *a*, β and ε .

Output: the inferred friend set $FIN_b(t)$, $F_b(t)$

- 1. **if** $FID_b(t) \neq \emptyset$
- 2. $m = \operatorname{nrow}(Ab) // \operatorname{get} \operatorname{number} \operatorname{of} \operatorname{distinct} \operatorname{nodes} \operatorname{in} Ab$
- 3. i = indexOf(Ab, t) // the index of t in Ab
- 4. $V_0 = \text{setInitialVector}(i, m)$
- 5. R = initR(i, m)
- 6. Pb = convertToTransitionMatrix(Ab)
- 7. *n*=1
- 8. $V_n = a \times Pb \times V_{n-1} + (1-a) \times R$
- 9. **while** $|V_n V_{n-1}| \ge \varepsilon$
- *10. n*++

$$V_n = a \times Pb \times V_{n-1} + (1-a) \times R$$

- 12. end while
- 13. // rank the probability from high to low
- 14. rankedProb = rank(V_n)
- 15. $k = |\beta \times m|$
- 16. s = 0;
- 17. **for** j = 1 to m **do**
- 18. f = getNode(rankedProb[j])
- 19. **if** $f \neq t$ and $f \notin FID_{h}(t)$
- 20. add f to $FIN_b(t)$
- 21. *s*++
- 22. end if
- 23. **if** s = k **do**
- 24. break
- 25. end if
- 26. **end for**
- 27. $F_b(t) = FID_b(t) \cup FIN_b(t)$
- 28. end if

Note that there could be incorrectly inferred friends in this step since the inferred friends actually have relatively higher probabilities to be friends of a target.

In the next section, we will demonstrate the effectiveness of the friend identification and the friend inference steps using various measures. In addition, we will discuss the parameters adopted in the algorithms and show their sensitivity for attack results.

4. EXPERIMENTS

In this section, we introduce the setup of the experiments for *FII* attacks and demonstrate the results of *FII* attacks. We also investigate the attack results in order to find what values of parameters and social features can promote the attacks.

4.1 Datasets

In this paper, we adopt three real social network datasets to demonstrate *FII* attacks:

- **D1**: It is a Facebook friend network dataset used in [9]. We remove the isolated nodes in the original dataset since they are useless in our experiments.
- **D2**: It is another Facebook friend network dataset adopted in [10]. Since it is not a well-connected network, we filter the nodes and choose the giant component of the original dataset for our experiments.
- **D3**: It is a Foursquare friend network dataset used in [11]. This network is a well-connected network.

The basic statistical information of these datasets is shown in Table 1 and the users' cumulative degree distributions are shown in Figure 5. We can see that users in D1 have the highest degrees and they are most connected with each other according to the average clustering coefficient and average path length. Users in D3 have the least degrees but they are more connected with each other than those in D2, although users in D2 have higher degrees than those in D3.

	D1	D2	D3
Nodes	3,963	63,392	10,326
Edges	88,156	816,886	52,974
Ave. Degree	44.490	25.773	10.260
Ave. Clustering Coefficient	0.617	0.253	0.285
Ave. Path Length	3.776	8.087	4.41
Triangles	1,612,010	3,501,534	99,334

Table 1. Basic Statistics Information of the Datasets



Figure 4. Degree Distributions of the Datasets

4.2 Setup of Experiments

4.2.1 Initialization of Adversaries and Targets

Based on the cumulative degree distributions in Figure 4, we first divide each dataset into three subsets with different ranges of degrees. The statistical information of these subsets is described in Table 2. We then randomly choose 100 adversaries from each of these subsets. These selected adversaries have diverse degrees and better represent various adversaries in these datasets. We finally get 300 adversaries in each dataset and 900 adversaries in total. These adversaries' degrees and their numbers of 2-distant neighbors are shown in Figure 5. Other social network features about the adversaries, such as the average clustering coefficient and average number of triangles are summarized in Table 3.



Figure 5. Degrees and the Numbers of 2-distant Neighbor of the Adversaries

From Figure 5, we can see that the number of 2-distant neighbors of an adversary is usually much larger than his degree in all datasets. In D2 and D3, the adversaries in the subset 1 have less 2distant neighbors than those in subset 3. However, in D1, the adversaries' average number of 2-distant neighbors in all the subsets seems to be equal. Specially, there is one adversary that his degree is larger than his 2-distant neighbors in D1.

From Table 3, we can see that adversaries in D1 are usually closer to other nodes since they have higher clustering coefficient values and lower the closeness centrality and the betweenness centrality values. We will analyze the relationships between these network features and the attack results later in this section.

	Range of the Node Degree in Subset 1	Range of the Node Degree in Subset 2	Range of the Node Degree in Subset 3
D1	(10, 30]	(30, 60]	> 60
D2	(8, 20]	(20, 40]	> 40
D3	(5,10]	(10, 20]	> 20

Table 2. Basic Statistical Information of the Subsets²

Table 5. Network Measures for the Auversaries						
	D1	D2	D3			
Ave. Clustering Coefficient	0.552	0.223	0.236			
Ave. Number of Triangles	1837.81	253.63	69.847			
Ave. Eccentricity	6.553	9.6	8.333			
Ave. Closeness Centrality	3.748	3.926	3.937			
Ave.						

Table 3. Network Measures for the Adversaries

For each selected adversary, we first randomly choose five friends as his targets. Then, we also randomly select five of the adversary's 2-distant neighbors as his additional targets. Hence, each adversary has 10 targets in total. In this paper, we call each pair of an adversary and a corresponding target as an attack instance. Thus, we have 3,000 attack instances on each dataset and we generate 9,000 attack instances in total. The basic statistical information of the chosen targets is shown in Table 4.

15945.516

Betweenness

Centrality

141809.841

35086.388

Table 4. Basic Statistical Information for the Targets

	Number of Attack Instances	Number of Targets ³	Ave. Degree of Targets	
D1	3,000	2020	54.282	
D2	3,000	2822	68.864	
D3	3,000	2224	21.760	

 $^{^{2}}$ In these three datasets, we do not consider users who have very few degrees (*e.g.*, less than 5 in D3) as adversaries.

In Theorem 2 in Section 3.2, we introduce $IS_b(t)$ that is the ideal exposed friends of an attack instance. Figure 6 shows the differences between t's degrees (|F(t)|) and the ideal exposed number of t's friends for b $(|IS_b(t)|)$. We can obviously find that b is ideally able to find more friends of t when b and t are friends. This is because the tendency of the curve expressing $IS_b(t)$ is very close to that representing F(t). When b and t are 2-distant neighbors, b may still identify and infer many friends of t in D1 but he may identify and infer relatively less friends of t in D2 and D3. The reason is that there are larger differences between F(t) and $IS_b(t)$ for many targets in D2 and D3. Hence, we expect that b has a higher probability to identify and infer more friends of t in D1 should be more successful than those in D2 and D3.



Figure 6. Targets' Degrees VS Ideal Numbers of Friends

4.2.2 Initialization of Parameters

When introducing the friend inference step in Section 3.3.2, we mentioned three parameters in the algorithm: a, β and ε . In all of our experiments, we set $\varepsilon = 10^{-6}$ for the equation $|V_n - V_{n-1}| < \varepsilon$. a and β are dynamically set in our experiments to demonstrate their impact on attack results.

4.3 Measurements

In our experiments, we use the following measures to evaluate the *FII* attacks:

• True Positive = $|F_b(t) \cap F(t)|$, which represents the number of correctly identified and inferred friends in an attack instance.

³ Different adversaries may have the same target in the attack instances.

When it is larger, it indicates that an adversary can find more of a target's friends.

- False Positive = $|F_b(t)| |F_b(t) \cap F(t)|$, which represents the incorrectly inferred friends in an attack instance. When it is smaller, it denotes that most of inferred friends for a target are accurate.
- Precision = $|F_b(t) \cap F(t)| / |F_b(t)|$, which represents the accuracy of an attack instance. When the precision is 1, it indicates that all the identified and inferred friends for *t* are correct. On the other hand, when it is 0, it represents all the identified and inferred friends are wrong.
- Recall = $|F_b(t) \cap F(t)| / |(F(b) \cup D_2(b)) \cap F(t)|$, which represents the percentage of correctly exposed friends by the ideal number of the exposed friends (refer to Theorem 2) in an attack instance. The higher the recall the more successful is an attack instance.
- Coverage = $|F_b(t) \cap F(t)| / |F(t)|$, which represents the percentage of correctly exposed friends for the target in an attack instance. A higher coverage value indicates that an adversary can find more percentage of a target's friends in an attack instance.
- F1 = 2×Precision×Recall / (Precision + Recall), which evaluate attack results by counting both the precision and the recall. F1 reaches its best value at 1 and worst score at 0.

Compared the recall with the coverage, the recall reflects the effectiveness of the attack algorithms while the coverage indicates the effectiveness of the *FII* attack.

Note that we do not present the time cost of an attack instance in our experiments in this paper since almost all attack instances can finish in 2 seconds. The *FII* attacks are efficient.

4.4 Experimental Results

4.4.1 Comparisons of Attack Results for Different Datasets

In the experiments presented in this subsection, we set a = 0.15 $[31]^4$ and $\beta = 0.01$ for all the attack instances conducted in D1, D2 and D3. We compare attack results on these datasets and try to identify: 1) whether attacks targeting friends of adversaries are more successful than those targeting 2-distant neighbors of adversaries; 2) what social features of the overall network can promote the attacks. Figures 7, 8 and 9 demonstrate the true positive, the false positive, the recall, the coverage and the precision for all attack instances on these three datasets. Note that Figure 7 ranks attack instances based on the increasing values of the true positive; Figure 8 ranks attack instances based on the decreasing values of the false positive; Figure 9 ranks attack instances based on the increasing values of the attack instance in *x* axis in different figures does not represent the same attack in our experiments.

From Figure 7, we can see that adversaries launching *FII* attacks can find more friends of the corresponding targets in D1 than in D2 and D3. For example, when adversaries and targets are friends, there are around 67% of attack instances in D1 where adversaries

can identify and correctly infer more than 20 friends of targets. However, there are few attack instances in D2 and D3 where adversaries can achieve the same goal. In addition, we can see that an adversary can find much more friends of a target via *FII* attacks when they are friends. When they are 2-distant neighbors, the adversary can only identify or correctly infer less than 20 friends of the target in most attack instances.



Figure 7. Comparisons of the True Positive among Datasets

Based on Figure 8, we can get that adversaries will get more incorrectly inferred friends for targets in D2 than in D1 and D3. The number of incorrectly inferred friends in D2 is more than 10 for around half of attack instances. Some attack instances in D2 can even get more than 30 incorrectly friends. However, we find that the relationships (friends or 2-distant neighbors) between adversaries and targets will not have significant impact on the false positive in most attack instances.



Figure 8. Comparisons of the False Positive among Datasets

Figure 9 has three graphs which show the recall, the coverage and the precisions in D1, D2 and D3. We can see that the values of the recall are relatively higher in D1 than those in D2 and D3. In D1, values of the recall are higher when adversaries and targets are friends than those when they are 2-distant neighbors. However, it is not the case in D2 and D3. We surprisingly find that the values of the recall seem a bit higher when adversaries and targets are 2-distant neighbors in many attack instances in D2 and D3.

The values of the recall and the coverage are almost same when adversaries and targets are friends on all the datasets. However, when they are 2-distang neighbors, the values of the coverage are much lower in some attack instances in D1 and in many attack instances in D2 and D3. Such a result is reasonable (refer to the definitions of the recall and the coverage in Section 4.3) since there are larger differences between targets' degrees and the set of the ideal exposed friends of the targets in many attack instance in D2 and D3, as shown in Figure 6.

The values of the precision are diverse according to the increase of the recall. Generally, when adversaries and targets are friends,

⁴ *a*=0.15 is the empirical value for the page rank using the random walk with restart [31].

most values of the precision are higher than 0.8 in D1; such values are usually less than 0.5 in D2; and a significant number of these values are equal to 0.5 in D3. When adversaries and targets are 2distant neighbors, the values of the precision are less than 0.3 in D1 and D2. However, many of these values are 0.5 in D3. It indicates that the relationships between adversaries and targets have relatively less impact on the values of the precision in D3. We also find that the values of the precision are usually larger than those of the recall in all the attack instances in Figure 9. It infers that it may be better to increase the number of inferred friends (increase β) to achieve a better recall and a not downgraded precision. We will demonstrate such experiments in Section 4.4.2.



Figure 9. Precision, Recall and Coverage in Datasets⁵

Based on above analysis, *FII* attacks in D1 should be most successful since adversaries from D1 can achieve a higher true positive, a lower false positive, a higher recall, a higher coverage and a higher precision. *FII* attacks in D3 are the second successful while those in D2 are least successful. By comparing the basic network statistics shown in Table 1, we may get the conclusion that a higher average clustering coefficient and a lower average path length of the entire network can promote *FII* attacks. Other features for the entire social graph, such as number of nodes, number of edges, the average degree of nodes and the number of

triangles, empirically, have insignificant impact on attack results. In theory, when the clustering coefficient is higher and the path length is smaller, it indicates that nodes in the graph tend to cluster together. If nodes tend to cluster in a social graph, it denotes that the set of ideal exposed friends will be larger and hence a *FII* attack has a higher possibility to be successful. This result suggests that an adversary should launch a *FII* attack in a more highly clustered network, in order to achieve a more successful attack.

4.4.2 Attack Results with Different a and β

In the experiments presented in this subsection, we conduct *FII* attacks only in D1. We try to find what values of the parameter *a* and β can promote the attack results.

First, we show the impact of *a* on attack results. Remind that *a* is the parameter to determine the probability that a node follows the random walk. [32] shows that a = 0.15 is not always the best value for the random walk with restart approach. Its value should depend on the diameter λ of a social graph. A = 0.15 works on the web graph where $\lambda = 19$. In our case, assume λ in an attack instance is round 3. Theoretically, we can get a better value of *a* should be 0.65 from $(1-0.15)^{19}=(1-a)^3$.

We design four sets of experiments with β =0.01. In each set of the experiments, we conduct 3,000 attack in D1 and the value of *a* in each attack instance is same. The values of *a* in these sets of experiments are 0.15, 0.65, 0.8 and 0.9, respectively. The attack results are shown in Table 5.

We can see that all the measures (precision, recall, coverage and F1) are improved with the increase of *a*. For example, when a=0.9, the average precision, recall and F1 score of the attacks, where adversaries and targets are friends, can increase to 0.839, 0.558 and 0.63, respectively. These values are a bit higher than those of the attack instances where a = 0.15. In addition, when adversaries and targets are friends, the improvement of attacks is more significant than those where adversaries and targets are 2-distant neighbors.

Next, we show the impact of β on attack results. Remind that β is the parameter to determine the number of inferred friends in a *FII* attack. Intuitively, the precision may be improved with the increase of β in *FII* attacks. However, the improvement of the precision may cause the decline of the recall [33].

To show β 's impact, we design five sets of experiments with a=0.15. In each set of the experiments, we also conduct 3,000 attack in D1 and the value of β in each attack instance is same. The values of β in these sets of experiments are 0.01, 0.015, 0.02, 0.025 and 0.03, respectively. The attack results of these experiments are shown in Table 6.

As expected, with the increase of β , the precision declines while the recall and coverage increase in all attack instances. However, we can find that the decrease of the precision is a bit larger than the increase of the recall, when the adversary and the target are friends. Hence, the F1 score declines in this case. In the attack instance where the adversary and the target are 2-distant neighbors, although the precision decreases a bit, the F1 score increases and hence the overall performance of the attack is promoted. Above results related to β suggest that an adversary can set a smaller value of β when he conducts attacks on his friends and set a larger value of β when he launches attacks on his 2distant neighbors, in order to achieve more successful attacks.

⁵ The blue cross in the figures represents the precision value of each attack instance.

β=0.01	b and t are friends				b an	d t are 2-dist	ant neighbor	S
	ave. precision	ave. recall	ave. coverage	ave. F1	ave. precision	ave. recall	ave. coverage	ave. F1
<i>a</i> =0.15	0.826	0.551	0.551	0.62	0.366	0.171	0.165	0.198
<i>a</i> =0.65	0.833	0.555	0.555	0.626	0.368	0.171	0.166	0.199
<i>a</i> =0.8	0.836	0.557	0.557	0.628	0.368	0.171	0.166	0.199
<i>a</i> =0.9	0.839	0.558	0.558	0.63	0.369	0.172	0.166	0.199

Table 5. Attack Results with a

Table 6. Attack Results with β

<i>a</i> =0.15	b and t are friends			b and t are 2-distant neighbors			rs	
	ave. precision	ave. recall	ave. coverage	ave. F1	ave. precision	ave. recall	ave. coverage	ave. F1
$\beta = 0.01$	0.826	0.551	0.551	0.62	0.366	0.171	0.165	0.198
$\beta = 0.015$	0.78	0.572	0.572	0.617	0.323	0.186	0.18	0.199
$\beta = 0.02$	0.741	0.592	0.592	0.613	0.296	0.202	0.195	0.2
$\beta = 0.025$	0.71	0.61	0.61	0.609	0.277	0.216	0.209	0.201
$\beta = 0.03$	0.683	0.626	0.626	0.605	0.264	0.229	0.221	0.203

4.4.3 Adversaries' Network Features and Attack Results

In theory, the result of a *FII* attack should depend on both an adversary and a target's social features. However, we assume that an adversary does not know any link information about a target initially in the attack. Hence, the adversary does not know the social feature of the target but he may know some of his social features. In this subsection, from perspective of an adversary, we try to identify what values of the adversary's social features can promote the attack.

In this subsection, we demonstrate *FII* attacks with a=0.9 and $\beta=0.02$ and conduct 3,000 attacks in D1. We will examine the relationships between attack results and an adversary's degree, his clustering coefficient, number of his 2-distant neighbors, number of mutual friends between him and a target⁶, number of triangles associated with him.



Figure 10. Impact of the Degree on Attack Results

The results are demonstrated from Figure 10 to Figure 14. In these figures, the values of the precision, recall and F1 are the average results for each adversary (remind that there are 10 targets for one adversary). Also note that it is possible that several adversaries have the same value for certain social features (*e.g.* node degree, number of 2-distant neighbors and *etc.*). Thus, in the figures, with a same *x*-coordinate value, there may be different *y*-coordinate values. For example, with x = 50 degree in Figure 10, there are two different values for the precision (0.45 and 0.5), recall (0.25 and 0.3) and F1 (0.32 and 0.38).

From Figure 10, we can find that when the degree is lower (less than 50), the values of the precision, recall and F1 are diverse. They can be low (less than 0.3) or high (more than 0.5). With the increase of the degree, we can see that the values of the precision, recall and F1 seem increase almost linearly. It indicates that an adversary who has many friends may have a higher probability to launch *FII* attacks more successfully.

⁶ It is actually equal to the result $FID_b(t)$ from the friend identification step.



Figure 11. Impact of the Number of 2-Distant Neighbor on Attack Results

In Figure 11, it seems that the overall trends of the precision, recall and F1 decline with the increase of the number of 2-distant neighbors an adversary has. It suggests that the number of 2-distant neighbors of an adversary should be limited in order to conduct *FII* attacks more successfully. Such a result makes sense since the random walk with restart algorithm may have a better result with a limited number of candidate nodes.

In Figure 12, the precision does not change significantly with the increase of the number of triangles. However, the trends of the recall and F1 look like a parabola opening downwards when the number of triangles are larger than 500. Initially, the values of the recall and F1 increase to the peaks (around 11,000) with the growing of the triangles. Then, the values of the recall and F1 decline with the continuous increase of the triangles. Such a result indicates that the increasing number of triangles cannot always promote the *FII* attacks. An adversary may have a medium number of triangles in order to launch a more successful *FII* attack.



Figure 12. Impact of the Number of Triangles on Attack Results

From Figure 13, we can see that the tendencies of the precision, recall and F1 seem rise with the increase of the values of the clustering coefficient when the values of the clustering coefficient are less than 0.4. After that the precision, recall and F1 do not change significantly. The overall tendencies of these measures tend to be logarithmic curves. It denotes that a higher value of the clustering coefficient of an adversary may promote *FII* attacks.



Figure 13. Impact of the Clustering Coefficient on Attack Results

From Figure 14, we can see that the average value of the precision seems to increase with the increase of the number of mutual friends between adversaries and targets. However, the tendencies of the recall and F1 may be expressed as a logarithmic curve with the increase of the mutual friends. These results suggest that it is better for an adversary to have more mutual friends with a target in order to conduct a *FII* attack more successfully.

We also investigate the relationships between attack results and the values of the adversary's eccentricity, closeness centrality, betweenness centrality⁷. However, the correlations between attack results and them are very inconsistent and unclear. Hence, we may conclude when an adversary has higher degree, less number of 2distant neighbors, a medium number of triangles, a higher clustering coefficient and more number of mutual friends between him and a target, he may conduct more successful *FII* attacks.



Figure 14. Impact of the Number of Mutual Friends on Attack Results

5. CONCLUSIONS

In this paper, we identify the threat of the inconsistency policies for a friendship link involving two users in undirected social networks. We propose that such a threat can be abused by an adversary to launch a *FII* attack to identify and infer friends of a target who conceal his friend list from the adversary. We demonstrate attacks using three real social network datasets. Our experimental results show that the *FII* attack is generally efficient and effective when adversaries and targets are friends or 2-distant

⁷ Based on the graph theory, an adversary's closeness and the betweeness centrality, even the eccentricity, should be somehow related to attack results.

neighbors. Our comprehensive analysis for attack results suggests that *FII* attacks are more successful in a more highly clustered network and a larger values of the parameters *a* and β can promote *FII* attacks. We also find that an adversary with a higher degree, a higher clustering coefficient value, more mutual friends between him and a target, etc., could has a higher probability to conduct more successful *FII* attacks.

In the future work, we will continue to propose more advanced *FII* attacks with more complex attack scenarios. For example, there are a number of friends (as targets) of an adversary who hide their friends from him. The adversary tries to identify and infer many friends of all these targets. In addition, we will also work on the resistant approach for *FII* attacks by decreasing the performances of the results.

6. REFERENCES

- Aleksandra Korolova, Rajeev Motwani, Shubha U. Nabar, and Ying Xu. Link privacy in social networks. In Proceeding of the 17th ACM conference on Information and knowledge management, 2008.
- [2] Daniel G. Avello. All liaisons are dangerous when all your friends are known to us. *In Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, 2011.
- [3] Arthur U. Asuncion and Michael T. Goodrich. Turning privacy leaks into floods: surreptitious discovery of social network friendships and other sensitive binary attribute vectors. In Proceedings of the 9th annual ACM workshop on Privacy in the electronic society, 2010.
- [4] Lei Jin, James B.D. Joshi and Mohd Anwar. Mutual-friend Based Attacks in Social Network Systems. *Computer & Security*, v. 37, pp 15-30, 2013.
- [5] Elena Zheleva and Lise Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles, *In Proceedings of the 18th international conference on World wide web*, 2009.
- [6] Zhen Wen and Ching-Yung Lin. On the quality of inferring interests from social neighbors. *In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010.
- [7] Leyla Bilge, Thorsten Strufe, Davide Balzarotti and Engin Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. *In Proceedings of the 18th international Conference on World Wide Web*, 2009.
- [8] Lei Jin, Hassan Takabi and James B.D. Joshi. Towards active detection of identity clone attacks on online social networks. *In Proceedings of the 1st ACM Conference in Data and Application Security and Privacy*, 2010.
- [9] Julian J. McAuley and Jure Leskovec. Learning to Discover Social Circles in Ego Networks. *In Proceedings of NIPS*, 2012.
- [10] Bimal Viswanath, Alan Mislove, Meeyoung Cha and Krishna P. Gummadi. On the evolution of user interaction in Facebook. In Proceedings of the 2nd ACM Workshop on online Social Networks, 2009.
- [11] Lei Jin, Xuelian Long and James Joshi. Towards understanding Residential Privacy by Analyzing User' Activities in Foursquare. In Proceedings of the 2012 ACM

Workshop on Building analysis datasets and gathering experience returns for security, 2012.

- [12] Zhijun Yin, Manish Gupta, Tim Weninger and Jiawei Han. A Unified Framework for Link Recommendation Using Random Walks. In Proceedings of 2010 International Conference on Advances in Social Networks Analysis and Mining, 2010.
- [13] Anastasios Noulas, Salvatore Scellato, Neal Lathia and Cecilia Mascolo. A Random Walk Around the City: New Venue Recommendation in Location-Based Social Networks. In Proceedings of 2012 International Conference on Social Computing, 2012.
- [14] Mark E. J. Newman. *Networks: An Introduction*. Oxford University Press, 2010.
- [15] Purnamrita Sarkar. Random Walk in Social Networks and their Applications: A Survey. *Social Network Data Analytics*, pp 43-77, 2011.
- [16] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. *Journal of the American Society* for Information Science and Technology, v. 58, pp 1019-1031, 2007.
- [17] Mohammad A. Hasan and Mohammed J. Zaki. A survey of link prediction in social net-works. *Social Network Data Analytics, pp 243–276.* 2011.
- [18] Zhijun, Yin, Manish Gupta, Tim Weninger, and Jiawei Han. LINKREC: a unified framework for link recommendation with user attributes and graph structure. *In Proceedings of the 19th international conference on World Wide Web*, 2010.
- [19] Neil Z. Gong, Ameet Talwalkar, Lester Mackey, Ling Huang, Eui C. R. Shin, Emil Stefanov, Elaine Shi and Dawn Song. Jointly Predicting Links and Inferring Attributes using a Social-Attribute Network (SAN). In Proceedings of the 6th SNA-KDD Workshop on Social Network Mining and Analysis, 2012.
- [20] [24] Suhendry Effendy, Roland H.C. Yap and Felix Halim. Revisiting Link Privacy in Social Networks. *In Proceedings* of the Second ACM conference in Data and Application Security and Privacy, 2012.
- [21] Joseph Bonneau, Jonathan Anderson, Ross Anderson and Frank Stajano. Eight Friends Are Enough: Social Graph Approximation via Public Listings. In Proceedings of the Second ACM EuroSys Workshop on Social Network Systems, 2009.
- [22] Vincent Leroy, B. Barla Cambazoglu and Francesco Bonchi. Cold Start Link Prediction. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, 2010.
- [23] Myunghwan Kim and Jure Leskovec. The Network Completion Problem: Inferring Missing Nodes and Edges in Networks. In Proceedings of the 11th SIAM International Conference in Data Mining, 2011.
- [24] Dóra Erdös, Rainer Gemulla and Evimaria Terzi. Reconstructing Graphs from Neighborhood Data. In Proceedings of the 12th International Conference in Data Mining, 2012.
- [25] Krishna P.N. Puttaswamy, Alessandra Sala and Ben Y. Zhao. StarClique: guaranteeing user privacy in social networks

against intersection attacks. In Proceedings of the 5th international conference on Emerging networking experiments and technologies, 2009.

- [26] Haiquan Chen, Wei-Shinn Ku, Haixun Wang, Liang Tang, and Min-Te Sun. LinkProbe: Probabilistic Inference on Large-Scale Social Networks. In Proceedings of the 29th IEEE International Conference in Data Engineering, 2013.
- [27] Cuneyt Gurcan Akcora, Barbara Carminati and Elena Ferrari. Privacy in Social Networks: How Risky is Your Social Graph. In Proceedings of the 28th IEEE International Conference in Data Engineering, 2012.
- [28] Cuneyt Gurcan Akcora, Barbara Carminati and Elena Ferrari. Risks of Friendships on Social Networks. In Proceedings of the 12th IEEE International Conference in Data Mining, 2012.
- [29] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, v. 390, pp 1150-1170, 2011.
- [30] Hanghang Tong, Christos Faloutsos and Jia-Yu Pan. Fast Random Walk with Restart and Its Applications. *In*

Proceedings of the Sixth International Conference in Data Mining, 2006.

- [31] Taher Haveliwala, Sepandar Kamvar and Glen Jeh. An analytical comparison of approaches to personalizing PageRank. Technical report, Stanford University, 2003.
- [32] Jia-Yu Pan, Hyung-Jeong Yang, Christos Faloutsos and Pinar Duygulu. Automatic Multimedia Cross-modal Correlatiin Discovery. In Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining, 2004.
- [33] Jesse Davis and Mark Goadrich. The Relationship Between Precision-Recall and ROC Curves. *In Proceedings of the* 23rd international conference on Machine learning, 2006.
- [34] Adam N. Joinson. Looking at, looking up or keeping up with people?: motives and use of facebook. In Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, 2008.
- [35] Danah Boyd and Eszter Hargittai. Facebook privacy settings: Who cares?. *First Monday*, vol. 15, no. 8, pp. 23, 2010.