A Two-stage Identification based on Seed and Grow Algorithm against Anonymized Social Networks

Mohammed Yaseen Navalur
M.Tech, Computer Network Engineering
T.John Institute Of Technology
Bangalore, India

M S. Nagashree K T
Assistant Professor, Dept of CSE
T.John Institute Of Technology
Bangalore, India

Abstract— **In a internet based social networking services the digital traces left behind even after anonymization they are more sensitive in privacy breaches. The sociologist tracks the feasibility of such an attack. There is an algorithm called Seed and Grow which identifies users from an anonymized social graph based only on graph structure. The algorithm first indentifies a seed sub graph either planted by the attacker or by any one ant then grows the seed layer based on the attackers existing knowledge of the users social relations. The algorithm identifies and relaxes implicit assumptions taken by previous work and improves the identification effectiveness and accuracy.**

Index Terms*— Feasibility, *Identification,Privacy and Accuracy,Graph.

I.INTRODUCTION

Social networks like Friendster.com, tagged.com, Xanga.com, LinkedIn including Facebook and Twitter, two popular online social networking services, rank at 2nd and 9th place respectively they have developed on the Internet over the past several years and these social networks have been successful in attracting many users, a decades ago only Telecommunication service providers and Intelligence agencies used to provide the critical information's like date of birth and other user generated contents, now through social networks the users engage with each other for various purposes, including business, entertainment and knowledge sharing. The commercial success of social networks depends on the number of users it attracts, and by encouraging users to add more users to their networks and to share data with other users in the social networks. End users are, however often not aware of the anonymiztion attacks and advertisements.

Due to the strong correlation between use data and the users sociality entity, privacy is a major concerning dealing with social network data in context such as storage, processing and publishing. Privacy control, through which a user can tune the visibility of her profile, is an essential feature in any major social networking service.

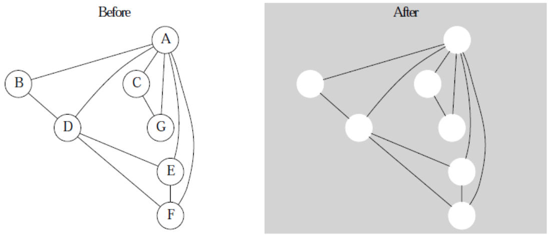


Figure 1.1 An illustration of naïve anonymization. Each node represents a user ID attached. Naïve anonymization simply removes the ID, but retains the network structure.

The common practice for privacy-sensitive social network data publishing is through anonymization,i.e., remove plainly identifying label such as name, social security number, post lore mail address, and retain the structure of the network as published data. Figure1.1 is as implemented illustration of this process. The motivation behind such processing prior to data publishing is that, by removing the "who" information, the utility of the social networks is maximally preserved with out compromising users' privacy. Narayanan and Shmatikov report several high profile cases in which "anonymity has been unquestioningly interpreted as equivalent to privacy" Can the aforementioned "naive" anonymization technique each I ever privacy preservation in the context to privacy-sensitive social network data publishing? This interesting and important question was posed only recently by Back strometal. A few privacy attacks have been proposed to circumvent the naïve anonymization protection..

Meanwhile, more sophisticated anonymization techniques have been proposed to provide better privacy protection [4, 5, 6, 7, 8]. Nevertheless, research in this area is still in its infancy and a lot of work, both in attacks and defenses, remains to be done.

In this paper, we describe a two-stage identification attack, *Seed-and-Grow*, against anonymized social net-works. The name suggests a metaphor for visualizing its structure and procedure. The attacker first plants a *seed* into the target social network before its release. After the anonymized data is published, the attacker retrieves the seed and makes it *grow* larger, thereby further breaching privacy.

More concretely, our contributions include:

- We propose an efficient seed construction and re-covery algorithm More specifically, we drop the assumption that the attacker has complete control over the connection between the seed and the rest of the graph (Section 3.1.2); the seed is constructed in a way which is only visible to the attacker (Section 3.1.2); the seed recovery algorithm examines at most the two-hop local neighborhood of each node, and thus is efficient (Section 3.1.3).
- We propose an algorithm which grows the seed (i.e., further identifies users and hence violates their privacy) by exploiting the overlapping user bases among social network services. Unlike pre-vious works which require arbitrary parameters for probing aggressiveness, our algorithm automat-ically finds a good balance between identification effectiveness and accuracy (Section 3.2).
- We demonstrate the significant improvements in identification effectiveness and accuracy of our algorithm over previous works with real-world social-network datasets.

II.RELATED WORK

A natural mathematical model to represent a social net-work is a graph. A graph G consists of a set V of vertices and a set $E \subseteq V \times V$ of edges. Labels can be attached to both vertices and edges to represent attributes.

In this context, *privacy* can be modeled as the knowl-edge of existence or absence of vertices, edges, or labels. An extension is to model privacy in terms of metrics, such as betweenness, closeness, and centrality, which originate from *social network analysis* studies [9].

The naive anonymization is to remove those labels which can be uniquely associated with one vertex (or a small group of vertices) from V . This is closely related to traditional anonymization techniques employed on rela-tional datasets [10]. However, the information conveyed in edges and its associated labels is susceptible to privacy breaches. Backstrom et al. [3] proposed an identification attack against anonymized graph, and coined the term *structural steganography*.

Besides privacy, other dimensions in formulating pri-vacy attacks against anonymized social networks, as identified in numerous previous works [5, 6, 8, 11], are the published data's *utility*, and the attacker's *background knowledge*.

Utility of published data measures information loss and distortion in the anonymization process. The more information that is lost or distorted, the less useful published data is. Existing anonymization schemes [4, 5, 6, 8, 11] are all based on the trade-off between the usefulness of the published data and the strength of protection. For example, Hay et al. [8] propose an anonymization algorithm in which the original social graph is partitioned into groups before publication, and "the number of nodes in each partition, along with the density of edges that exist within and across partitions," are published.

Although a trade-off between utility and privacy is necessary, it is hard, if not impossible, to find a proper balance overall. Besides, it is hard to prevent attackers from proactively collecting intelligence on the social net-work. It is especially relevant today as major online so-cial networking services provide APIs to facilitate third-party application development. These programming in-terfaces can be abused by a malicious party to gather information about the network.

Background knowledge characterizes the information in the attacker's possession which can be used to compro-mise privacy protection. It is closely related to what is perceived as privacy in a particular context.

The attacker's background knowledge is not restricted to the target's neighborhood in a single network, but may span multiple networks and include the target's alter egos in all of these networks [2]. This is a real-istic assumption. Consider the status quo in the social networking service business, in which service providers, like Facebook and Flickr, offer complementary services. It is very likely that a user of one service would simul-taneously use another service. As a person registers to different social networking services, her connections in these services, which relate to her social relationships in the real world, might reveal valuable information which the attacker can make use of to threaten her privacy.

The above observation inspires Seed-and-Grow, which exploits the increasingly overlapping user-bases among social networking services. A concrete example is helpful in understanding this idea.

[Motivating scenario.] Bob, as an employee of a social networking service provider F-net, acquires from his employer a social-network data-set, in which vertices represent users and edges represent private chat sessions. The edges are labeled with attributes such as timestamps. In accordance with its privacy policy, F-net has removed the user IDs from the graph before giving it to Bob.

Bob, being an inquisitive person, wants to know who these users are. Suppose, somehow, Bob iden-tifies 4 of these users from the graph (the "Seed Construction" and "Seed Recovery" interludes in Section 3.1 illustrate a way to do this). By using a graph (with the user ID tagged) he crawled a month ago from the website of another service provider T-net (the 4 identified persons are also users of T-net), and by carefully measuring structural similarity of these graphs, he manages to identify 100 more users from the anonymized graph (the "Dissimilarity" in-terlude in Section 3.2 illustrates a way to do this). By doing so, Bob circumvents his employer's attempt to protect its customers' privacy.

We conclude this section with a brief comment on our choice of model. We use the *undirected* graph model to explain the proposed deanonymization attack on social networks. Undirected graphs arise naturally in scenarios where the social relation under investigation is *mutual*, e.g., friend requests must be confirmed on Facebook. *Directed* graphs, however, are more appropriate in other cases, such as fans following a movie star on Twitter. An undirected graph could be seen as a special case of directed graphs, in which the relationship is reciprocal; Mislove et al. confirmed the relationship reciprocity in a large-scale study on the Flickr online photo-sharing service [12]. As explained in Section 3, the algorithms used in the proposed deanonymization attack do not rely on the fact that the used graphs are undirected; they work on directed graphs the same way. The undirected graph model is only a choice for specificity and ease of presentation.

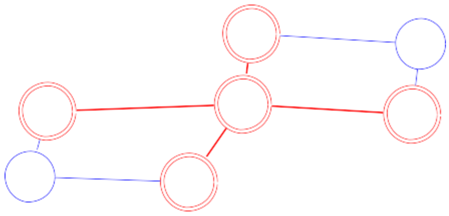
III.SYSTEM MODEL

SEED-AND-GROW: THE ATTACK

This section describes an attack that identifies users from an anonymized social graph. Let an undirected graph $G_T = \{V_T , E_T \}$ represent the *target* social network after anonymization. We assume that the attacker has an undirected graph $G_B = \{V_B , E_B \}$ which models his *background knowledge* about the social relationships among a group of people, i.e., V_B are labeled with the identities of these people. The motivating scenario demonstrates one way to obtain G_B .

The attack concerned here is to infer the identities of the vertices V_T by considering *structural similarity* between the target graph G_T and the background graph G_B : Nodes that belong to the same users are assumed to have similar connections in G_T and G_B . Although sporadic connections between who would otherwise be strangers may exist in an online social network (and, thus, affect the similarity between G_T and G_B), such links can be removed by, for example, quantifying the strength of these connections [13]; the residual network consists of the stable, strong connections that reflect the users' real-world social relationships, which give rise to the similarity between G_T and G_B . Additionally, auxiliary knowledge about the target graph G_T (such as the source and nature of the graph) may help in choosing a background graph G_B with similar structures.

Thus, the two graphs G_T and G_B are syntactically (the social connections) similar but semantically (the meaning associated with such connections) different. By re-identifying the vertices in G_T with the help of G_B , the attacker associates the sensitive semantics with users on the anonymized G_T and, thus, compromise the privacy of such users. An example of sensitive semantics is the private chat sessions, and their associated timestamps, in the motivating scenario.



•

Figure3.1.ArandomlygeneratedgraphG_F maybesymmetric. Verticesin$G_F = \{v_1,...,v_5\}$aredouble-circled.

We assume that, *before* the release of G_T , the attacker obtains (either by creating or stealing) a few accounts and connects them with a few other users (the *initial seeds*) in G_T . The feasibility of doing this is the basis of the Sybil identity forgery attack studied in numerous previous works [14, 15, 16, 17, 18, 19, 20, 21, 22]. In-deed, experiments (Section 4) show that our algorithm is capable of identifying 10 times of anonymized users from as few as 5 initial seeds. Besides user IDs, the attacker knows nothing about the relationship between the initial seeds and other users in G_T . Furthermore, unlike previous works, we *do not assume that the attacker has complete control over the connections*: the attack only *knows* them before G_T 's release. This is more realistic. An example is a confirmation-based social network, in which a connection is established only if the two parties confirm it: the attacker *can decline but not impose* a connection.

In contrast to a pure structure-based vertex matching algorithm [23], Seed-and-Grow is a *two-stage* algorithm.

The *seed* stage plants (by obtaining accounts and establishing relationships) a small specially designed sub-graph $G_F = \{V_F , E_F \} \subseteq G_T$ (G_F reads as "fingerprint") into G_T before its release. After the anonymized graph is released, the attacker locates G_F in G_T . The neighboring vertices V_S of G_F in G_T are readily identified and serve as the *initial seeds* to be grown.

The *grow* stage is essentially comprised of a structure-based vertex matching, which further identifies vertices adjacent to the initial seeds V_S . This is a self-reinforcing process, in which the seeds grow larger as more vertices are identified.

3.1 Seed

3.1.1 Feasibility

Successful retrieval of G_F from G_T is guaranteed if G_F exhibits the following structural properties.

- G_F is *uniquely* identifiable, i.e., no subgraph $H \subseteq G_T$ except G_F is isomorphic to G_F . For example, in Figure 2, sub graph $\{v_1, v_2, v_3\}$ is isomorphic to sub graph

{v_1, v_4, v_5} because there is a structure-preserving mapping v_1 7→ v_1, v_2 7→ v_4, v_3 7→ v_5 between them. Therefore, the two sub graphs are structurally indistinguishable once the vertex labels are removed.

if we could locate VF={v1,….v5}from GT, v2,..v5 are indistinguishable once their labels are even

In practice, since the structure of other nodes in the network is unknown to the attacker before its release, the uniquely identifiable property is not realizable. How-ever, as was proved by Backstrom et al. [3], with a large enough size and randomly generated edges under the Erdos¨-Renyi´ model [24], G_F will be uniquely identifiable with high probability.

Although a randomly generated graph G_F is very likely to be uniquely identifiable in G_T , it may violate the asymmetric structural property. However, because the goal of seed is to identify the initial seed V_S rather than the fingerprint G_F , the asymmetric requirement for G_F can be relaxed. For u ∈ V_S , let V_F (u) be the vertices in V_F which connect with u (|V_F (u)| ≥ 1 by the definition of V_S). For each pair of vertices, say u and v, in V_S , as long as V_F (u) and V_F (v) are distinguishable in G_F (e.g., |V_F (u)| 6= |V_F (v)| or the degree sequences are different; more precisely, no automorphism of G_F exists which maps V_F (u) to V_F (v)), and once G_F is recovered from G_T , V_S can be identified uniquely. In Figure 2, since V_F (6) = {v_2, v_3} and V_F (7) = {v_4, v_5} are not distinguishable, vertices v_6 and v_7 cannot be identified through G_F .

Based on these observations, we propose the following method of constructing and recovering G_F .

3.1.2 Construction

The construction of G_F starts with a *star* structure. The motivation for the star structure will become clear in Section 3.1.3. We call the vertex at the center of the star the *head* of G_F and denote it by v_h. v_h connects and *only* connects to every other vertex in G_F .

The vertices in V_F − {v_h} are connected with some other vertices of the initial seeds V_S in G_T . To ensure the distinguish ability of two seeds u and v once the fingerprint G_F is recovered, the attacker can decline those connection requests (from other vertices in G_T) which render V_F (u) = V_F (v). Note that the attacker is not assumed to have full control over the connections: an attacker does not have to impose a connection as long as he can decline it.

- G_F is *asymmetric*, i.e., G_F does not have any non-

trivial automorphism. For example, in Figure 2, sub-graph {v_1, v_2, . . . , v_5} has an automorphism v_1 7→ v_1, v_2 7→v_3, v_3 7→v_4, v_4 7→v_5, v_5 7→v_2. Therefore,

After setting up the initial star structure, the attacker establishes other *internal* connections within the finger-print graph G_F . Two principles dictate this process:

1) No automorphism of G_F should map V_F (u) to V_F (v) for two distinct initial seeds u and v.
2) The constructed G_F should leave no distinctive structural pattern for anyone besides the attacker, but should yet be recoverable.

Principle 1 follows from the discussion in Section 3.1.1: a pair of initial seeds u and v could be unambiguously identified only if no automorphism of G_F maps V_F (u) to V_F (v). Principle 2 apparently presents a dilemma: G_F should mingle with the rest of the target graph G_T , yet be distinctive. In the following discussion, we first justify this principle, and then resolve the dilemma by reconciling the two competing requirements.

The motivation for having G_F mingle with the rest of the target graph G_T is to avoid leaving distinctive structural patterns for defenders. Otherwise, a straight-forward defense against the proposed attack would be to locate the fingerprint graph G_F by pattern-matching and to remove it prior to the publication of G_T . An implication is that the construction of G_F should be stochastic rather than deterministic.

Yet, stochastic construction alone is not enough for G_F to blend into G_T . Numerous studies [25, 26, 27, 28, 29, 30, 31] indicate the existence of distinctive structural properties of online social networks as opposed to arbitrary random graphs. In particular, online social graphs consist of a well-connected backbone linking numerous small communities [25]. Within each community, vertices show a local, transitive, triangle-closing connection pat-tern [29]. The construction of G_F should reflect these properties to blend into G_T.

The cost for the attacker to set up the fingerprint graph G_F is dominated by the number and variety of connections between V_F and the initial seeds V_S . To minimize the cost, the construction of G_F mimics a local com-munity in G_T [25]: after establishing the star structure centering at the head vertex v_h, each pair of vertices in V_F −{v_h} connects with a probability of t. The probability t reflects the *transitivity* of a community in G_T , which is the likelihood that, in the same community, two vertices sharing a common neighbor (v_h in G_F) will connect to each other. In reality, the attacker almost always knows some auxiliary information about the target graph G_F , which may include the community transitivity and a reasonable size for a community: The construction of G_F should be adjusted to such information for G_F to blend into G_T

After connecting pairs of non-head vertices in V_F with a probability of the community transitivity t, the attacker collects the *internal degree* D_F (v), which is number of vertices in V_F that v connects to, for every v ∈ V_F − {v_h} into an *ordered* sequence S_D.

Now, for every v ∈ V_S , v has a corresponding subsequence S_D(v) of S_D according to its connectivity with V_F . For example, in Figure 2, v_6 connects to v_2 and v_3 from G_F ; since D_F (v_2) = D_F (v_3) = 1, S_D(v_6) = h1, 1i. As long as S_D(u) 6= S_D(v) for u and v from V_S , no automorphism of G_F will map V_F (u) to V_F (v). There-fore, the attacker guarantees unambiguous recovery of V_S by ensuring that the randomly connected G_F satisfies this condition. If not, the attacker will simply redo the random connection among V_F −{v_h} until it does (which it eventually will, since we assume that V_F (u) 6= V_F (v) for any pair u and v from V_S). Algorithm 1 summarizes the procedure.[Seed construction.] Bob had created 7 accounts v_h and v_1, . . . , v_6, i.e., V_F . He first connected v_h with v_1, . . . , v_6. After a while, he noticed that users

v_7 to v_{10} are connected with v_1, . . . , v_6, i.e., V_S = {v_7, . . . , v_{10}}.

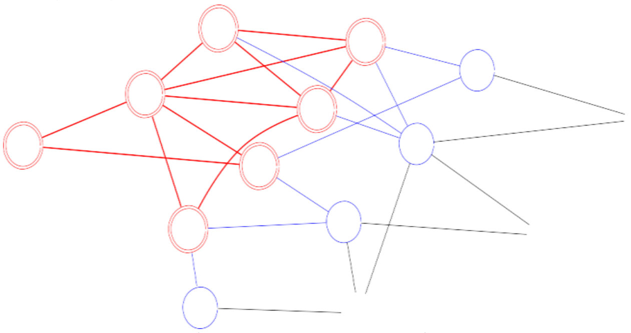


Fig. 3.2 The task of the seed stage is to identify the initial seed by recovering the fingerprint graph G_F.

Then, he randomly connected v_1, . . . , v_6 with the community transitivity t and got the resulting graph G_F ,as shown in Figure 3. The ordered internal

degree sequence S_D = h2, 2, 2, 3, 3, 4i.Bob found out that S_D(v_7) = h2i, S_D(v_8) = h2, 2i, S_D(v_9) = h3, 3, 4i, and S_D(v_{10}) = h2, 3i. Since they are mutually distinct, Bob was sure that he could

identify v_7 to v_{10} once V_F was recovered from the published anonymized graph.

3.2 Grow

The initial seeds V_S provide a firm ground for further identification in the anonymized graph G_T. Background knowledge G_B comes into play at this stage.

We have a partial mapping between G_T and G_B , i.e., the initial seeds V_S in G_T map to corresponding vertices in G_B . Two examples of partial graph mappings are the Twitter and Flickr datasets [2] and the Netflix and IMDB datasets [32]. The straightforward idea of testing all possible mappings for the rest of the vertices has an exponential complexity, which is unacceptable even for a medium-sized network. Besides, the overlapping

Figure 3.3 shows a small example. v_7 to v_{10} have already been identified in the seed stage (recall Figure 3). The task is to identify other vertices in the target graph G_T .

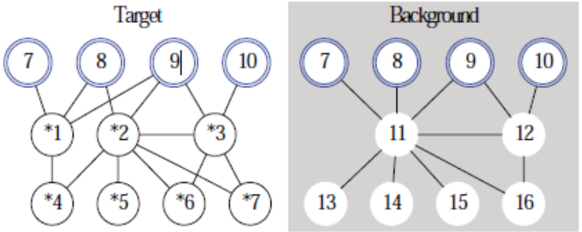


Figure 3.3 The task of the grow stage is to identify the unmapped vertices starting from the seed.

based on the attacker's existing knowledge of the users' social relations. We identify and relax implicit assump-tions for unambiguous seed identification taken by pre-vious works, eliminate arbitrary parameters in grow algorithm, and demonstrate the superior performance over previous works in terms of identification effective-ness and accuracy by simulations on real-world-collected social-network datasets

between G_T and G_B may be *partial*, so a *full* mapping is neither possible nor necessarily desirable. Therefore, the grow algorithm adopts a progressive and self-reinforcing strategy, starting with the initial seeds and extending the mapping to other vertices for each round

IV.CONCLUSION

Seed-and-Grow, to identify users from an anonymized social graph. Our algorithm exploits the increasing overlapping user-bases among services and is based solely on social graph structure. The algorithm first identifies a seed sub-graph, either planted by an attacker or divulged by collusion of a small group of users, and then grows the seed larger

REFERENCE

[1] B. Krishnamurthy and C. E. Wills, “Characterizing privacy in online social networks,” in *Proc. ACM WOSN*, 2008.

[2] A. Narayanan and V. Shmatikov, “De-anonymizing social net-works,” in *Proc. IEEE S&P*, 2009.

[3] L. Backstrom, C. Dwork, and J. Kleinberg, “Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography,” in *Proc. ACM WWW*, 2007.

[4] M. Hay, G. Makalu, D. Jensen, P. Weis, and S. Srivastava, “Anonymizing social networks,” Univ. Massachusetts, Amherst, Tech. Rep., 2007.

[5] E. Zheleva and L. Getoor, “Preserving the privacy of sensitive relationships in graph data,” in *Proc. ACM SIGKDD*, 2007.

[6] A. Korolova, R. Motwani, S. Nabar, and Y. Xu, “Link privacy in social networks,” in *Proc. ACM CIKM*, 2008.

[7] B. Zhou and J. Pei, “Preserving privacy in social networks against neighborhood attacks,” in *Proc. Intl. Conf. on Data Engineering (ICDE)*. IEEE, 2008.