http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (19): 4882-4888, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.4882.4888

De-Anonymization of Dynamic Social Networks

Xuan Ding, Lan Zhang, Zhiguo Wan and Ming Gu
School of Software, Tsinghua University, Beijing, 100084, China

Abstract: Nowadays, online social network data are being increasingly published to third parties. It has been shown that individually sensitive information can be recovered from the released data and several anonymization techniques have been proposed. However, most of these defenses have focused on "one-time" releases and do not take into account the re-publication of dynamic social network data. Re-publishing data periodically is a natural result of social network evolution and an emerging requirement of dynamic social network analysis. In this paper, we show that by utilizing correlations between sequential releases, the adversary can achieve high precision in de-anonymization of the released data, suppressing the uncertainty of re-identifying each release separately and synthesizing the results afterwards. Besides, we combine structural knowledge with node attributes to compromise graph modification based defenses. With experiments on real data, this work is the first to demonstrate feasibility of de-anonymizing dynamic social networks and should arouse concern for future works on privacy preservation in social network data publishing.

Key words: Social network analysis, privacy, de-anonymization

INTRODUCTION

Over the past few years, the rise of online social networks in popularity has produced large quantities of data and attracted dramatic interest from the literature. While social network data are valuable to sociologists, economists, data-mining researchers and many others, their containing sensitive information of individuals have aroused serious privacy concerns. Recent studies have shown that network structure can be used to re-identify individuals from the released data. For example, in (Backstrom *et al*., 2007), individuals are re-identified from the released network owing to their unique connection patterns to an embedded subgraph that the adversary is aware of; and in (Narayanan and Shmatikov, 2009), identities of users are exposed because of their alike network structure between the released network and the adversary's auxiliary network.

These existing attacks all use some sort of background knowledge to map from individuals with known identities to anonymized nodes in the released network. For example, in (Narayanan and Shmatikov, 2009), the adversary exploits an external, auxiliary network that overlaps with the released network as its background knowledge to re-identify nodes. Successful re-identification of anonymized nodes would expose potentially sensitive information (e.g. attributes, relationships) of them, breaching their privacy. However, in practice, the lack of ground truth (i.e. the true mapping between the adversary's knowledge and the released network) often lowers down the reliability of re-identification result.

As social network evolves and subsequent snapshots of the same network are re-published for evolutionary and dynamic analysis, the impact of lacking ground truth becomes more serious. In appearance, re-publication of the same network could provide a chance of breaching more to the adversary, who just needs to re-identify each release separately and add up the results. However, in contrast to this intuition, if the adversary is to synthesize results with false mappings, it will end up with a combination in which nodes representing different individuals are mapped to be the same, and/or nodes representing the same individual are mapped to be different.

As an example, we assume that the adversary tried to map Alice from its background knowledge, S_1, to two successively released snapshots, S_2 and S_3 and the ground truth to be ⟨Alice, a, u⟩, i.e., Alice was anonymized to be node a in S_2 and u in S_3. Table 1 shows the result of re-identifying Alice in S_2 and S_3 separately, in which P(x) indicates the possibility that Alice maps to node x. We can notice from this table that Alice was falsely mapped to v in S_3, resulting in incorrect combination ⟨Alice, a, v⟩. However, if the adversary utilized the correlations

Table 1: De-anonymizing releases separately

S_1	S_2	S_3	Result
Alice	P(a) = 0.6	P(u) = 0.4	P(a, v) = 0.36
	P(b) = 0.4	P(v) = 0.6	⟨Alice, a, v⟩

Corresponding Author: Xuan Ding, School of Software, Tsinghua University, Beijing, 100084, China

between S_2 and S_3 and found out that, for example, there is high confidence that a and u represent the same individual, it would correctly reveal ⟨Alice, a, u⟩ with high possibility.

Besides this ambiguity brought by dynamics, another limitation of the existing attacks is that they exploit only structural knowledge for re-identification. In fact, ever since the attack of (Backstrom *et al.*, 2007), many defenses have been proposed to protect anonymized nodes from being re-identified by various kinds of structural knowledge (Zhou and Pei, 2008). These methods ensure that adversaries holding only structural knowledge cannot re-identify any node with a probability higher than 1/k.

To address these challenges, first develop a technique called threading to utilize correlations between releases. Instead of de-anonymizing each release separately and synthesizing the resulting mappings afterwards, we de-anonymize all releases simultaneously and ensure unambiguity of the result all through the process. Secondly, to compromise recent defenses, we exploit both structural knowledge and node attributes to re-identify anonymized nodes. Both techniques improve the reliability of result, as they produce more accurate mappings between the adversary' knowledge and the released networks.

In brief, we make the following contributions:

- This is the first work to de-anonymize dynamic social network releases. Feasibility of our method is demonstrated with real data
- Although primitive and intuitive, this is the first attempt to combine structural knowledge with node attributes to de-anonymize social network data
- We proposed the concept of threading to associate time-varying sets of elements. We believe this concept can help in solving some analogous problems

RELATED WORK

Social network de-anonymization: Backstrom et al. are among the first to demonstrate feasibility of de-anonymizing social network data (Backstrom *et al.*, 2007). They proposed a family of attacks, i.e., the active, the passive and the semi-passive attacks, to breach edge privacy of a targeted group of individuals. All of these attacks share the same the philosophy: within the released network, the adversary first locates a seed network (denoted as H) that it has detailed degree and internal structure knowledge of and, then, by utilizing knowledge of the unique connection patterns between H and the targets, the adversary re-identifies each target and learns whether edges exist or not between each pair of them.

The attacks of (Backstrom *et al.*, 2007) are limited to privacy breach of only a small set of targets. Basing on this small, seed mapping as a starting point, Narayanan *et al.* developed an algorithm to extend the seed mapping iteratively to a larger mapping by utilizing knowledge of an external, auxiliary network that overlaps with the released network on more nodes in addition to those in *H* (Narayanan and Shmatikov, 2009). In each iteration, the existing mapping (initially, the seed mapping) is extended to a number of neighbors according to structural similarity and forms a larger mapping, which, will be fed back to the next iteration if the convergence criteria has not been met.

Hay *et al.* (2008) also described a series of structural attacks, in which the adversary knows the local structure, embedded subgraph, or connection patterns to network hubs of the target. All of the above attacks are based on structural knowledge only and are not specially designed for de-anonymizing dynamic social network releases. As we have already noted, the attack of (Narayanan and Shmatikov, 2009) suffers from the ambiguity in combining separate mappings. And although less ambiguous, the attacks of (Backstrom *et al.*, 2007) rely strongly on the assumption that the internal structure of the seed network stays exactly the same in anonymization, which can hardly be the case under recent defenses. The same assumption is implicated in the seed identification stage in Backstrom *et al.* (2007). In contrast, our method employs the "threading" technique to deal with dynamics and introduces node attributes into re-identification to loosen the very assumption.

Another category of attacks rely more on node attributes than on structural knowledge. Wondracek *et al.* (2010), present an attack that exploits group membership information on social networking sites for re-identification. Specifically, it is shown that group memberships of a user can serve as its "fingerprint" in the data and be used to re-identify it or, at least, to reduce the number of candidates. This attack has its similarity to that of (Narayanan and Shmatikov, 2008), in which an anonymized dataset containing movie ratings about thousands of movies from 500,000 people is de-anonymized under the fact that the possibility of two individuals to have identical ratings on the same subset of movies is extremely low.

In brief, both (Wondracek *et al.*, 2010) and (Narayanan and Shmatikov, 2008) are based on the high dimensionality and sparsity of node attributes in certain networks. While node attributes are also exploited for re-identification in our algorithm, we make no assumptions on distributions of them.

Social network anonymization: On defenses, existing works have focused mostly on privacy-preservation of "one-time" releases (Wu *et al.*, 2010). Zhou et al. categorized these works into clustering-based and graph modification approaches (Zhou *et al.*, 2008). Briefly, a clustering-based method hides details of individuals by clustering corresponding nodes and edges into super-nodes and super-edges (Hay *et al.*, 2008; Campan and Truta, 2008; Bhagat *et al.*, 2009), while a graph modification method provides anonymity by modifying nodes and edges to be less outstanding (Liu and Terzi, 2008; Zhou *et al.*, 2008; Zou *et al.*, 2009). There are also other anonymization methods for "one-time" releases (Wu *et al.*, 2010). Due to space limitations, we refer interested readers to (Zhou *et al.*, 2008) and (Wu *et al.*, 2010) for more details.

Recently, the problem of anonymizing periodically re-published social network data to support dynamic analysis has aroused interest in the literature. Anonymization in such a setting is challenging, as an adversary can collect historical information and use it for re-identification. Zou *et al.* (2009), described a technique called ID generalization to handle this problem, in which node IDs are replaced with ID sets (called generalized ID) to ensure the number of candidates against structural queries. To allow dynamic data analysis, this technique preserves the original ID in generalized ID in all releases. In our case, however, this just makes finding correlations between releases more efficient.

MODEL AND DEFINITIONS

In this section we introduce notations, definitions and preliminary facts that are used throughout this paper. Particularly, we propose "threading", one of the most important concepts of this paper and a number of useful facts relating to it.

Dynamic social network: A dynamic social network is a social network that varies with time. A snapshot of a dynamic social network at a certain point in time, say t, is represented with an attributed graph $G_t = (V_t, E_t, X_t, Y_t)$, where V_t is a set of nodes representing participating entities, $E_t \subseteq [V_t]^2$ is a set of edges representing their connections (e.g. relationships, interactions) and X_t and Y_t are sets of attributes attached to nodes in V_t and edges in E_t, respectively.

This notation is mostly from (Narayanan and Shmatikov, 2009), except that we take dynamics into consideration and do not explicitly specify the graph to be directed or undirected. Besides, we emphasize that the graph representing a social network is attributed, which is critical to our algorithm. For convenience, we adopt the following conventions on attributes as well:

- Given a node $v \in V_t$ (an edge $e \in E_t$), the notation $X_t[v]$ ($Y_t[e]$) represents the set of attributes attached to v (e)
- given an attribute $X \in X_t$ ($Y \in Y_t$), the notation X[v] (Y[e]) represents the value of the attribute X of v (Y of e)

Sequential releases: With the above notation, a series of n snapshots of a dynamic social network at time $t_1, t_2, ..., t_n$ are then represented with $G_{t1}, G_{t2}, ..., G_{tn}$, or $G_1, G_2, ..., G_n$ for short if the context is clear.

After anonymization, a network G_t is released as $G_t^* = (V_t^*, E_t^*, X_t^* \cup \{ID_t\}, Y_t^*)$. The released set of nodes V_t* does not have to be a subset of Vt; and if not, we assume that (V_t*-V_t) is a set of dummy nodes created only to satisfy certain anonymity criteria (e.g. k-automorphism (Zou *et al.*, 2009)). The released set of edges E_t^* may not be a subset of E_t neither. In this case, (E_t*-E_t) is assumed to be a set of dummy edges. We denote the attribute representing anonymized identifiers of nodes by ID_t. For convenience, we abbreviate ID_t [v] to v.ID_t in the rest of this paper; and if t is easily inferable from the context, we omit it and write v.ID for short.

Anonymized identifiers of the same entity in sequential releases may or may not stay the same. If the adopted anonymization method is not specially designed to support dynamic analysis, it is normal to preserve identifiers to allow such a purpose. In this case, there exist strong correlations between releases that can be easily exploited by an adversary. However, if dynamics is specially handled, for example, if the ID generalization technique (Zou *et al.*, 2009) is used, anonymized identifiers will vary and the correlations may be weakened. We regard it as a key issue to capture the various possibilities that would emerge of anonymized identifiers of the same entity. Let $v_{t1} \in V_{t1}^*$ and $v_{t2} \in V_{t2}^*$ be two nodes representing the same entity in two releases G_{t1}^* and G_{t2}^*. With their anonymized identifiers, v_{t1}. ID and v_{t2}. ID, we consider the following cases:

- Node identifiers are re-generated in each release and v_{t1}.ID and v_{t2}.ID tend to be distinct
- Node identifiers are retained for purpose of dynamic analysis and v_{t1}. ID = v_{t2}.ID is guaranteed
- Node identifiers are generalized for anonymity (Zou *et al.*, 2009) and v_{t1}.ID and v_{t2}.ID may not be the same. However, v_{t1}. ID n v_{t2}.ID $\neq \emptyset$ is guaranteed. (Note that ID is a set of identifiers in this case)

Further, for simplicity and to focus on the main objective of this work, i.e. the de-anonymization of dynamic releases, we assume in our model that node attributes are only sanitized but not modified by any means (e.g. perturbation, generalization). This indicates that for any node attribute to release, it is either removed, or reserved in the released network. Formally, we have $\forall v \in V_t^*$, $X_t^*[v] \subseteq X_t[v]$. We argue that this assumption can be removed by mending the matching strategy (Section 4), but we choose to stick to it as it yields a more concentrated algorithm. Edge attributes are not restricted by this assumption.

Threat and privacy model: Similar to all the existing attacks, we assume that the adversary has access to some kind of background knowledge for re-identification. Particularly, in our case, we assume the adversary to hold an auxiliary network that overlaps with every targeted release. This assumption is no stronger than that of (Narayanan and Shmatikov, 2009) since all the releases are inherently overlapped and an earlier crawled subgraph from the targeted network is sufficient to serve the purpose. We also assume that the adversary holds detailed knowledge of a small set of targets (i.e. the seed network), which can be used to re-identify them from each release. The same assumption is adopted in Backstrom *et al.* (2007) and Narayanan and Shmatikov (2009).

The purpose of de-anonymizing social network data is to re-identify individuals and breach their privacy, i.e., revealing information such as sensitive attributes and relationships that are not included in the background knowledge. In a dynamic setting, dynamic privacy is also considered. For example, a relationship establishment or break up between two individuals may be exposed if they are both re-identified.

Threading: As mentioned in Section 1, de-anonymizing each release separately and synthesizing the results afterwards do not de-anonymize the entire serials. In contrast, it brings more ambiguity into the final result and makes each mapping more unreliable. We introduce the concept of threading to formalize this problem.

Definition 1 (Thread/Threading Instance): Let $S_1, S_2, ..., S_n$ be n non-empty sets and $e_1, e_2, ..., e_n$ be n elements from each set, i.e. $e_i \in S_i$, i = 1, 2,..., n. The sequence $\langle e_1, e_2,..., e_n\rangle$ is said to be an n-dimensional thread (or threading instance) through $S_1, S_2, ..., S_n$.

Definition 2 (Threading): Let $S_1, S_2, ..., S_n$ be n non- empty sets and T = $\{\langle e^j, e^j,..., e^j\rangle j = 1, 2, ..., m\}$ be a set of m threads through $S_1, S_2,..., S_{n.}$ T is said to be a threading through $S_1, S_2, ..., S_n$ if and only if all of the m threads are disjoint, i.e. $\forall i \in \{1, 2,..., n\}$, $a, b \in \{1, 2,..., m\}$, $e^a = e^b \Leftrightarrow a = b$.

Table 2: Threading example

S_1	S_2	S_3	Threading instance
Alice	a	u	⟨Alice, a, u⟩
Bob	b	v	⟨Bob, b, v⟩
Charlie	c	w	⟨Charlie, c, w⟩

Table 2 gives an example of threading, in which S1 = {Alice, Bob, Charlie}, S_2 = {a, b, c} and S_3 = {u, v, w} are three non-empty sets and t_1 = ⟨Alice, a, u⟩, t_2 = ⟨Bob, b, v⟩, t_3 = ⟨Charlie, c, w⟩ are disjoint threads through them. The set T = $\{t_1, t_2, t_3\}$ is then a threading through S_1, S_2 and S_3

Definition 3 (Graph threading): Let $G_1, G_2,..., G_n$ be n graphs and $V_1, V_2, ..., V_n$ be their corresponding sets of nodes. T is said to be a graph threading through $G_1, G_2, ..., G_n$ if and only if T is a threading through $V_1, V_2,..., V_n$.

Threading extends the concept of one-to-one mapping (or mapping for short). However, we can always split a threading into mappings. For example, the 3-dimensional threading in Table 2 can be viewed to consist of two mappings from S_1 to S_2 and from S_2 to S_3 This concept is captured by Definition 4 and used in Algorithm 2.

Definition 4 (Induced Mapping): Let T = $\{\langle e_1^j, e_2^j,... e_n^j\rangle | j = 1, 2,..., m\}$ be a threading through n non-empty sets $^S{}_1, {}^S{}_2,..., {}^S{}_n$ and f: $\{e_a^j\} \rightarrow \{e_b^j\}$ be a mapping from $\{e_a^j\}$ to $\{e_b^j\}$ for some $a, b \in \{1, 2, ..., n\}$, $a \neq b$. f is said to be the induced mapping of T from S_a to S_b if and only if $\forall j \in \{1, 2,..., m\}$, $f(e_a^j) = e_b^j$.

Algorithm 1: Seed Threading

```
Input: The seed network s and the released networks $g_1, g_2,..., g_n$.
Output: The seed threading $t_{sd}$.
$t_{sd} \leftarrow \varnothing$
foreach $g_i$ in $\{g_1, g_2, ..., g_n\}$ do
map s to $g_i$ and obtain mapping $m_i$
foreach v in s.nodes do
insert $\langle v, m_1(v), m_2(v),..., m_n(v)\rangle$ into $t_{sd}$
return $t_{sd}$
```

Algorithm 2: Threading expansion

```
Input: The auxiliary network aux, the released networks $g_1, g_2, ..., g_n$
and the seed threading $t_{sd}$.
Output: The expanded threading $t_{ex}$
$t_{ex} \leftarrow t_{sd}$, $g_0 \leftarrow$ aux, convergence $\leftarrow$ false
 While not convergence do
 Convergence $\leftarrow$ true
 Foreach $v_0$ in $g_0$.nodes and not in $t_{ex}.m_0$ do
    for i $\leftarrow$ 0 to n - 1 do
     $v_{i+1} \leftarrow$ BestMatch ($g_i, g_{i+1}, t_{ex}.m_i, v_i$)
     if $v_{i+1}$ = None then
       continue the outer loop with the
next $v_0$
$v_{n+1} \leftarrow$ BestMatch ($g_n, g_0, t_{ex}$.mn, $v_n$)
if $v_{n+1} = v_0$ then
insert $\langle v_0, v_1, v_2, ..., v_n\rangle$ into $t_{ex}$
convergence $\leftarrow$ false
return $t_{ex}$
```

Theorem 1: Let S_1, S_2, ..., S_n be n non-empty sets and E_1, E_2, ..., E_n be n non-empty subsets of each set, i.e. $E_i \subseteq E_i$, i = 1, 2,..., n. If $|E_1| = |E_2| = ... = |E_n|$ and $\{f_i: E_i \rightarrow E_{i+1}\}$ are (n-1) mappings between each pair of E_i and E_{i+1}, i = 1, 2,..., (n-1), letting $f_1^* = f_1$, $f_{i+1}^* = f_{i+1}(f_i^*)$, then $T = \{\langle e, f_1^*(e), f_2^*(e), ..., f_{n-1}^*(e)\rangle | e \in E_1\}$ is a threading through S_1, S_2, ..., S_n.

Proof 1: Omitted due to space limitations.

DE-ANONYMIZATION

Our re-identification algorithm runs in a similar two-stages manner to (Narayanan and Shmatikov, 2009). It begins with the construction of a seed threading through the adversary's seed network (which is part of the complete but less detailed, auxiliary network) and the released networks. Once the seed threading is constructed, the algorithm advances to the main, threading expansion stage, in which the seed threading is extended iteratively to its neighborhood to eventually a large threading through the auxiliary and the released networks. The details of the algorithm are as follows.

Seed threading: As mentioned earlier (Section 1), exploiting only structural knowledge is not sufficient to re-identify nodes under recent defenses. Therefore, we extended the algorithm of (Backstrom *et al*., 2007) and (Narayanan and Shmatikov, 2009) to also exploit node attributes in re-identification. Specifically, besides the degree and internal structure tests, a third, attribute test is performed to rule out candidates that are structurally similar but with distinct attributes. This modification is straightforward and, due to space limitations, we just omit the details here.

```
Algorithm 3: BestMatch
    Input: lgraph, rgraph, mapping, lnode.
    Output: The node rnode in rgraph that best matches the node lnode in
    lgraph; or None if such a node does not exist
    Foreach rnode in rgraph.nodes do
    Scores[rnode] $\leftarrow$MatchingScore (lnode, rnode)
    If Eccentricity(scores) >= theta then
    Return rnode with the maximum score
    else
    Return None
```

With this new algorithm, we are able to locate the seed nodes within each release and combine the resulting mappings together to obtain the seed threading through the seed network and the released networks. Note that the combination here is reliable because re-identification of seed nodes is assumed to be accurate (Section 3). Algorithm 1 summarizes and describes this process. The resulting t_{sd} is a graph threading through s, g_1, g_2,..., g_n And since s is a subgraph of aux, t_{sd} is also a graph threading through aux, g_1, g_2,..., g_n (by Theorem 1).

Threading expansion: After the construction of seed threading, we use it as a starting point to expand to large threadings with the aid of the complete, auxiliary network. The idea of expanding seed threading iteratively to its neighborhood to obtain an eventual, large threading is not new (Narayanan and Shmatikov, 2009). However, our primary contribution here lies in the threading of multiple networks simultaneously, emitting the ambiguity of re-identifying each network separately; and lies in the combinative exploitation of node attributes with structural knowledge in re-identification, breaking more existing defenses.

The expansion algorithm is given in Algorithm 2. It takes as input the networks to thread, i.e. the auxiliary network aux and the released networks g_1, g_2,..., g_n and the seed threading t_{sd} to expand. Note that $t_{ex}.m_i$ denotes the induced mapping from g_i to g_{i+1}, i = 0, 1, ..., (n-1) and $t_{ex}.m_n$ is the induced mapping from g_n to g_0. The expansion stops when there is no new thread to be inserted into t_{ex}.

Matching score computation: The procedure BestMatch (Algorithm 3) is the key for re-identification to succeed. Taken as input two networks lgraph and rgraph, a bijection mapping in between and a node lnode in lgraph, this algorithm tries to find the node rnode in rgraph that best matches lnode based on mapping.

As we exploit both structural knowledge and node attributes for re-identification, computation of matching score is divided into two parts accordingly. The structural part, Ms, is calculated in the same way as (Narayanan and Shmatikov, 2009). Let V_L be the set of lnode's neighbors that have images in mapping and V_R be the set of V_L's images that are rnode's neighbors, then:

$$MS = \sum_{v \in V_R} \frac{1}{\sqrt{\Delta_v}}$$

And the attribute part, M_a, is calculated with:

$$M_a = \frac{\left|\left\{X \in X_t^* \,|\, X[\ln ode] = X[rnode]\right\}\right|}{\left|X_t^*\right|}$$

Then the overall matching score is:

$$M_{overall} = \alpha \cdot M_S + \beta \cdot M_a$$

where α and β are weighting variables.

EXPERIMENTS

We used data crawled from Netease Microblog, one of the top social networking sites in China, to evaluate our algorithm. As shown in Table 3, Netease Microblog was

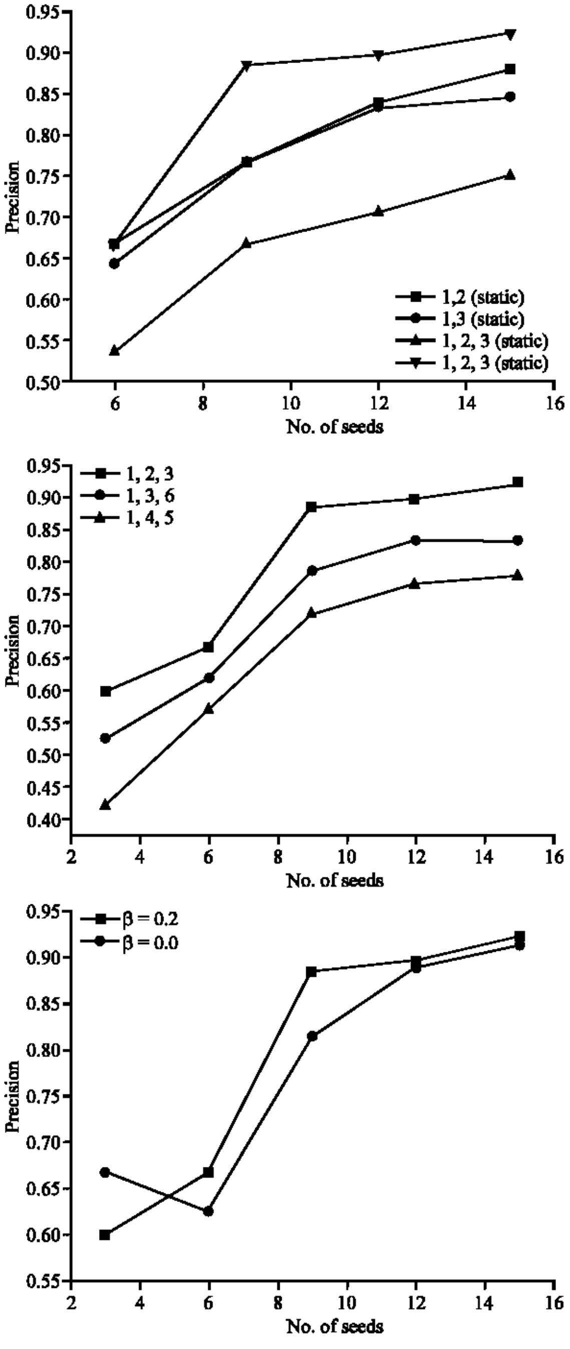

Fig. 1: De-anonymization results on real data (from Netease Microblog). (a) Separate vs. threading-based attacks (b) Re-identification of different release combinations and (c) Effect of node attributes

experiencing a rapid growth in users during Mar. 27th and Apr. 16th in 2010, which makes it a good source of dynamic data for our evaluation.

For simplicity, we extracted the 1-neighborhood of a randomly selected node, v, from S_1 as the adversary's seed network and the 2-neighborhood to be the complete auxiliary network (edges between nodes that are both not v's neighbors are removed). Sequential releases are obtained by anonymizing the other snapshots, i.e. S_2 to S_6.

Table 3: Data crawled from netease microblog

No.	Date	Nodes	Edges	Av. deg
S1	2010-03-27	47,367	784,508	33.1
S2	2010-03-30	54,190	858,244	31.7
S3	2010-04-02	64,002	977,889	30.6
S4	2010-04-07	117,864	1,478,188	25.1
S5	2010-04-11	161,267	1,894,840	23.5
S6	2010-04-16	204,790	2,308,505	22.5

We first compared our algorithm with (Narayanan and Shmatikov, 2009) and summarize the results in Fig. 1a, from which we can see clearly that de-anonymizing releases separately (line "1,2" and "1,3") and synthesizing the results afterwards do lower down the reliability of result significantly; and our threading-based technique can yield result of much higher precision. Second, we evaluated our algorithm against different combinations of releases, i.e. S_2 and S_3, S_4 and S_5 and S_3 and S_6, to validate its general effectiveness and examine the affect of time to re-identification. The results are shown in Fig. 1b. As we can see, the algorithm yields better threading when the releases are close in time (line "1,2,3"), especially when they are close to the background knowledge (line "1,3,6" vs. "1,4,5"). And unsurprisingly, the algorithm yields result of high precision when the size of the seed network is large. Figure 1c shows the effect of node attributes to re-identification by comparing a zero and nonzero value of â in matching score computation.

CONCLUSION

In this study, we demonstrated the feasibility of utilizing correlations between sequential social network releases to achieve high precision in de-anonymization of the released data. We exploited both structural knowledge and node attributes in our algorithm to re-identify anonymized nodes. This combination not only helps in finding reliable correlations and leading to results of high precision, but also compromises most existing graph modification based defenses as they take only network structure into consideration. Real data experiments showed the effectiveness and superiority of our method compared to the existing attacks.

REFERENCES

Backstrom, L., C. Dwork and J. Kleinberg, 2007. Wherefore art thou r3579x? anonymized social networks, hidden patterns and structural steganography. Proceedings of the 16th International Conference on World Wide Web, May 8-12, 2007, Banff, Alberta, Canada, pp: 181-190.

Bhagat, S., G. Cormode, B. Krishnamurthy and D. Srivastava, 2009. Class-based graph anonymization for social network data. Proceedings of the VLDB Endowment, Volume 2, August 2009, Lyon, France, pp: 766-777.

Campan, A. and T.M. Truta, 2008. A clustering approach for data and structural anonymity in social networks. Proceedings of the 2nd ACM SIGKDD International Work shop on Privacy, Security and Trust in KDD, August 24, 2008, Las Vegas, Nevada..

Hay, M., G. Miklau, D. Jensen, D. Towsley and P. Weis, 2008. Resisting structural reidentification in anonymized social networks. Proceedings of the VLDB Endowment, Volume 1, August 23-28, 2008, Auckland, New Zealand, pp: 102-114.

Liu, K. and E. Terzi, 2008. Towards identity anonymization on graphs. Proceedings of the ACM SIGMOD International Conference on Management of Data, June 9-12, 2008, Vancouver, BC., Canada, pp: 93-106.

Narayanan, A. and V. Shmatikov, 2008. Robust de-anonymization of large sparse datasets. Proceedings of the IEEE Symposium on Security and Privacy, May 18-22, 2008, Oakland, CA., pp: 111-125.

Narayanan, A. and V. Shmatikov, 2009. De-anonymizing social networks. Proceedings of the 30th IEEE Symposium on Security and Privacy, May 17-20, 2009, Berkeley, CA., pp: 173-187.

Wondracek, G., T. Holz, E. Kirda and C. Kruegel, 2010. A practical attack to de-anonymize social network users. Proceedings of the IEEE Symposium on Security and Privacy, May 16-19, 2010, Oakland, CA, USA., pp: 223-238.

Wu, X., X. Ying, K. Liu and L. Chen, 2010. A survey of privacy-preservation of graphs and social networks. Adv. Database Syst., 40: 421-453.

Zhou, B. and J. Pei, 2008. Preserving privacy in social networks against neighborhood attacks. Proceedings of the 24th IEEE International Conference on Data Engineering, April 7-12, 2008, Cancun, pp: 506-515.

Zhou, B., J. Pei and W.S. Luk, 2008. A brief survey on anonymization techniques for privacy preserving publishing of social network data. ACM SIGKDD Explo Rations Newslett., 10: 12-22.

Zou, L., L. Chen and M.T. Ozsu, 2009. K-automorphism: A general framework for privacy preserving network publication. Proceedings of the VLDB Endowment, Volume 2, August 24-28, 2009, Lyon, France, PP: 946-957.