# SA Framework based De-anonymization of Social Networks

Honglu Jiang; Jiguo Yu, *IEEE Memeber*; Chunqiang Hu, *IEEE Member*;
Xiuzhen Cheng, *IEEE Fellow*; Cheng Zhang

**Abstract**—The data of social networks contains a large amount of personal information, of which may be public or insignificant, but some may be sensitive and private. Once the user privacy leaked, it may bring a variety of troubles for users. To protect the privacy of users, holders of social networking data first conduct anonymization before the data is published. However, the simple anonymity method does not play a very good protection. At present, a number of de-anonymization attacks for the data releasing of social networks have arisen. These de-anonymization attacks are mostly based on the network structure with the use of feature matching and other methods. In this paper, we integrate the structural characteristics of social networks and the attributes of social nodes, thus the social network is modeled as a Structure-Attribute (SA) framework. The similarity measurement of social network nodes is proposed, with the consideration of the structure similarity and attribute similarity. The accuracy of node matching and de-anonymization is improved. We also present the similarity analysis of different measurements and discuss how graph anonymization techniques affect network characteristics with elaborate experimental results. We conduct our de-anonymization based on three realistic datasets to verify the accuracy and efficiency.

**Index Terms**—social network; privacy preservation; $k$-anonymity; de-anonymization; differential privacy.

## 1 Introduction

A large number of different online social networks have gained tremendous popularity in recent years. These social networks provide the online information sharing and exchange platform for different groups with different functions [1]. As a true social portrayal, social networking data contains a lot of personal information. In many social networking sites, users will be asked to fill in personal information such as name, gender, birthday, educational level, work situation, marital status, e-mail or even personal photos. In addition, the text, pictures, videos, geographical location published by users (User-generated content) will also be retained in the database. According to the privacy policy of social networking sites and users' privacy preferences, some of these information may be open or insignificant, and some may be very sensitive and private [2]. Once the user privacy leaked, it may bring troubles to the users, such as receiving spam mail, junk message or telephone harassment. It may also cause damages to the personal reputation, property damage, and even brings personal injury [3]. So the privacy issue has received a lot of attention and has been paid much attention. For example, the nature of smart homes inevitably raises security and privacy concerns [4]. Due to the close correlation with individuals physical features and status, the adoption of Cyber-Physical Social Systems (CPSSs) has been inevitably hindered by users' privacy concerns [5]. The Internet of Things (IoT) changes human lives greatly by connecting every objects together. As a result, billing can be conducted automaticly from the shopping cart itself, preventing customers from waiting in a long queue at checkout. Therefore, it is necessary to design secure communication protocols to make the system practical [6].

To protect the involved user privacy, holders of social networking data will be anonymized before the data is released [7], [8]. In general, data anonymization techniques can be divided into three categories: naive ID removal, $k$-anonymization (randomly increase or delete a few edges) [9], [10], and differential privacy [11], [12].

However, the naive ID removal does not play a very good protection effect. There have been a number of de-anonymization attacks to the releasing data of social networks [13], [14], [15], [16], [17], [18], [19], [20]. These technologies are mostly based on the network structure, with the use of node degree or connectivity and other information. It also conducts the disclosure of identity of anonymous users with the adoption of feature matching and other means. Based on a variety of background knowledge, the de-anonymization has a strong attack ability. The naive ID removal has been proven that it cannot resist the de-anonymization attack. For the $k$-anonymity method, it is not well protected against structurally anonymous attacks. Differential privacy was originally intended to protect the privacy of interactive queries. However, structure-based de-anonymization attacks are generally non-interactive queries.

In the existing de-anonymization attacks, the accuracy of the de-anonymization result is unknown because of the

- *H. Jiang is with the School of Information Science and Engineering, Qufu Normal University, Rizhao, 276826, P.R. China. She is also with Shandong Polytechnic, Jinan, 250104, P.R. China. E-mail: jianghonglu88@163.com*
  *J. Yu (Corresponding Author) is with the School of Information Science and Engineering, Qufu Normal University, Rizhao, 276826, P.R. China. E-mail: jiguoyu@sina.com*
  *C. Hu is with the College of Software Engineering, Chongqing University, Chongqing, 400044, P.R. China. E-mail:chu@cqu.edu.cn*
  *X. Cheng and C. Zhang are with the Department of Computer Science, The George Washington University, Washington DC 20052. E-mail: cheng@gwu.edu, zhangchengcarl@gwu.edu*

*Manuscript received ; revised*

inaccuracy of the feature matching process, and the absence of the true mapping from the attacker's background knowledge to the target network.

- In this paper, we model the social network as a Structure-Attribute (SA) framework which integrates the structural characteristics of social networks and social node properties, adding some attribute nodes to the social network. Each attribute node represents the user's attribute value. The link between the social user and the attribute node indicates that the user has the attribute value.
- We propose a novel similarity measurement of social network nodes, with the consideration of structural similarity and attribute similarity of social network nodes, so as to improve the accuracy of node matching and the accuracy of de-anonymization.
- We firstly adopt the spectral graph partition algorithm to partition large social networks into a number of small graphs. Then we develop a concrete de-anonymization algorithm including two phases. In the first phase, we construct a bipartite graph based on the anonymized graph and the auxiliary graph. In the second phase, the node matching problem is reduced to find a maximum weighted matching of the bipartite graph.
- We present the similarity analysis of different measurements and discuss how graph anonymization techniques affect network characteristics with elaborate experimental results. It is conducted on the Twitter dataset based on three anonymization methods.
- We simulate the de-anonymization attack on three realistic datasets. The experiment proves that the method in this paper can improve the accuracy of de-anonymization. Moreover, the run time and accuracy of the de-anonymization method are tested.

The remainder of this paper is organized as follows. In Section 2, we describe and analyze the methods of de-anonymization in recent years. The problem definition and network model are introduced in Section 3. We also present the definition and the hypothesis of the de-anonymization method, and introduce the SA framework model in detail. In section 4, we introduce the spectrum partitioning algorithm briefly to partition the large social network. Section 5 describes the definition of node similarity and the de-anonymization algorithm. In Section 6, we present the similarity analysis of different measurements and discuss the effect of graph anonymization techniques on network characteristics with many experimental results based on Twitter dataset. We collect three datasets to evaluate our de-anonymization attack in Section 7. The social structure and user attributes are collected to construct the SA framework. In Section 8, the de-anonymization attack is conducted on three social network datasets. The experiments prove the good performance of the accuracy and the efficiency of our de-anonymization algorithm. Section 9 summarizes our work and gives the prospect of further research works.

## 2 Related Work

The current privacy protection technology can be divided into three main categories: naive ID removal, $k$- anonymization [9], [10] and differential privacy protection [11], [12]. The naive ID removal is not well protected against de-anonymization attacks [13], [14], [18]. However, the naive ID removal is still widely used to anonymize data before data sharing, data publishing and/or data transferring. $k$-anonymization is an important way to protect information privacy when data is published. It requires that the published data contains a certain number (at least $k$) indistinguishable records so that the attacker cannot distinguish the specific privacy information belonging to individuals, thus protecting the privacy of individuals. $k$-anonymization protects the privacy of the individual to a certain extent, but it cannot resist the de-anonymization attack of node matching based on structural features. In [21], K. Xing *et al.* considered the problem of mutual privacy-protection in social participatory sensing, in which individuals contribute their private information to build a (virtual) community. Xing and Hu provided a mutual privacy preserving k-means clustering scheme which neither leaks individual's private information nor discloses the community's characteristic data. Differential privacy was first proposed in the traditional database area, and it has been widely used in various traditional data analysis tasks before applying to social networking data. By adding carefully calculated random noise to the query results, differential privacy ensures that any change in the set of records in the database does not statistically distinguish between query results. For social networking data, differential privacy is only applied to the simple statistical analysis of attribute information, such as the distribution of node degree, attribute value distribution [22], [23]. Afterwards, differential privacy began to be used to analyze the social network structure information, such as the data privacy protection in interactive queries [11], [12]. The differential privacy for social networks can be divided into node-based differential privacy and edge-based differential privacy.

At present, the inference attacks in social networks are divided into two main categories: private attribute inference [24], [25], [26], [27], [28], [29], [30], [31] and user de-anonymization [14], [18], [19], [20]. Private attribute inference intends to dig out the hidden attribute information that are intentionally protected by the users or data publisher. Attackers can easily collect information from social networks through some crawlers, which can be combined with other side channel information for the attribute inference. User de-anonymization attack adopts two graphs as input, with one anonymized and the other (the auxiliary graph) including the true user identities, and the purpose is to map the nodes in these two graphs so as to achieve the de-anonymization.

In [13], structure based de-anonymization was introduced where they proposed both active attacks and passive attacks to de-anonymize social network data. The basic idea of these attacks is to create a subgraph and a link pattern to the target users before releasing data. Then the target users can be de-anonymized by identifying the previously created subgraph and the link pattern. However, these attacks are not scalable and difficult to control for the continuous growth of social network data during the data release process. And for the passive attacks [13], it can be easily defended against by obfuscating the social network

structure, while it is still difficult to be extended to large scale social networks. In [14], Narayanan and Shmatikov *et al.* proposed a robust and scalable de-anonymization attack which is extended to large-scale directed social networks. This algorithm consists of two processes: seed identification and propagation. In the first phase, a set of seed mappings is identified. In the second phase, the de-anonymization is propagated from the seed mappings to other users in the anonymized graph with the adoption of several de-anonymization heuristics.

Srivatsa and Hicks *et al.* first proposed a de-anonymization attack to mobility traces, using social networks as the side-channel information [18]. They proposed three two-phase schemes to perform the de-anonymization. Firstly, a contact graph is constructed based on a mobility trace. Then, a social graph is adopted to de-anonymize the target contact graph subsequently. However, scalability is still an important limitation of this algorithm [18]. In [32], Ji *et al.* designed an adaptive de-anonymization (ADA) framework for the scenario that the anonymized and auxiliary graphs have partial overlap. ADA also includes seed identification phase and a propagation phase. In [33], Ji *et al.* presented the de-anonymization method under the configuration model. Moreover, a practical optimization-based de-anonymization algorithm is proposed. Nilizadeh *et al.* proposed a community-based de-anonymization scheme of social networks, which can be employed to enhance seed-based attacks [20]. In this scheme, community-level de-anonymization is first used. Subsequently, within each de-anonymization community, the obtained information can be used to enhance the user-level de-anonymization. In [19], Ji *et al.* conduct the first perfect de-anonymizability and partial de-anonymizability with seed information in general scenarios. In this de-anonymization scheme, social networks can follow an arbitrary distribution model. The detailed theoretical analysis for the existing structure based de-anonymization attacks is proposed in this quantification. In [34], Qian *et al.* introduced a knowledge graph to explicitly represent the prior information of the attacker for any individual user. Based on the defined knowledge graph, they formulate the process of de-anonymization and privacy inference. Ji *et al.* studied the impacts of non-Personal Identifiable Information on the privacy of graph data with attribute information in [35]. The attribute-based anonymity analysis for structure-attribute graph data are conducted.

## 3 Definition and Model

### Social Network

To make the paper more readable, all the notations are summarized in Table 1.

Assume that the attacker has a certain background knowledge to help them complete the de-anonymization. We use undirected graphs $G_s = (V_s, E_s)$ to represent social networks, where $E_s$ represents the social relationships between nodes in $V_s$. In addition to the social network structure, each node contains the associated attributes and behaviors. For instance, in Sina microblog set, nodes are the Sina microblog users, and edges represent the friendship between users. Node attributes can be derived from the user profile information such as age, gender, major, occupation, residence, etc. There are several kinds of behaviors in social networks, like giving comments, clicking "$Like$" button, forwarding other people's status or the set of items (apps, books). In this paper, we regard the user's behavior as an attribute.

Specifically, we need to distinguish between attributes and attribute values. For example, major, occupation, residence are different attributes, and each attribute may have many multiple attribute values; for example, the user's major can be computer science, mathematics or physics. Therefore, we adopt a binary representation for each attribute value, and the number of all the distinct attribute values are denoted as $d$. Then, the attribute information of the node $u$ can be represented as a $d$-dimensional binary column vector $\overrightarrow{b_u}$. The $i$th entry equals to 1 when node $u$ has the $i$th attribute value, and the value is 0 when the node does not have this attribute value. Therefore, the attribute values of all social nodes are represented by matrix $B = [\overrightarrow{b_1}, \overrightarrow{b_2} \dots \overrightarrow{b_{n_s}}]$, of which $n_s$ is the number of all social nodes.

### Social-Attribute Framework

Once a Social-Attribute (SA) network model is constructed, a social structure and user attributes are combined into a unified framework. Given a social network $G_s = (V_s, E_s)$ with an attribute matrix $B = [\overrightarrow{b_1}, \overrightarrow{b_2} \dots \overrightarrow{b_{n_s}}]$, an enhanced network is constructed by adding additional $d$ attribute nodes, where each node corresponds to an attribute value. For each node $u$ in $V_s$, when $u$ has the attribute value, the node $u$ has an undirected edge with the additional attribute node. Therefore, in the SA framework, the corresponding node in $G_s$ is called the social node; the node which represents the attribute value is called attribute node. The edge between the social nodes is called social link, and the edge between the social node and the attribute node is called attribute link. Fig.1 shows an example of a simple SA network. Bob, Alice, Linda, and Mike are social nodes; while Male, female, computer science, doctor, biology, age $< 30$ and angry birds are the attribute nodes. The friendship between Mike and Linda is considered as the social link, and the attribute link between Alice and computer science means that Alice has this attribute value.

### Network Model

**Anonymized graph.** We adopt the SA framework to model the social network. So the anonymized social network is modeled by graph $G^a = (V^a, E^a, t^a)$, where $V^a$ is the set of nodes, and $E^a$ represents all the links between the nodes, that is $E^a = \{e^a_{i,j} | i, j \in V^a, a\ social\ tie\ exists\ between\ i\ and\ j\}$, and $m^a = |E^a|$ represents the number of edges. Then, $t^a$ represents the node type. For instance, $t^a_i$ represents the type of node $i$, when $t^a_i = S$, it means that node $i$ is a social node, and when $t^a_i = A$, it means node $i$ is an attribute node. Given $\forall i \in V$, its neighborhood is defined as $N^a(i) = \{j \in V^a | e^a_{i,j} \in E^a\}$. Then we define $\Delta^a_i = |N^a(i)|$ as the number of neighbor nodes of $i$. The attribute information of the node $i$ can be expressed as a $d$-dimension binary vector $\overrightarrow{b^a_i}$, thus the matrix $B^a$ representing all attribute values of all social nodes.

TABLE 1
Description of Notations

| Variable Name | Description |
|---|---|
| $G_s = (V_s, E_s)$ | the social network |
| $B = [\overrightarrow{b_1}, \overrightarrow{b_2} \dots \overrightarrow{b_{n_s}}]$ | the attribute values of all social nodes |
| $G^a, G^u$ | the anonymized and auxiliary graph |
| $V^a, V^u$ | the node set in anonymized and auxiliary graph |
| $E^a, E^u$ | all the links between nodes in anonymized and auxiliary graph |
| $t_i^a, t_j^u$ | the type of node $i$ in anonymized graph and node $j$ in auxiliary graph |
| $N_i^a, N_j^u$ | the neighborhood of node $i$ in anonymized graph and node $j$ in auxiliary graph |
| $\sigma$ | the mapping of $V^a \rightarrow V^u$ |
| $\Delta_i^a, \Delta_j^u$ | the number of neighbor nodes of node $i$ in anonymized graph and node $j$ in auxiliary graph |
| $(\Delta_i^a)^S, (\Delta_j^u)^S$ | the number of social node neighbors of node $i$ in anonymized graph and node $j$ in auxiliary graph |
| $\overrightarrow{b_i^a}, \overrightarrow{b_j^u}$ | the attribute value of node $i$ in anonymized graph and node $j$ in auxiliary graph |
| $S_A(i,j)$ | the attribute similarity between node $i$ and $j$ |
| $\overrightarrow{A_i}, \overrightarrow{B_j}$ | the degree sequence of social neighbors of node $i$ in $G^a$and $j$ in $G^u$ |
| $S_R(i,j)$ | the structural similarity between node $i$ and $j$ |
| $S(i,j)$ | the similarity between node $i$ and $j$ |
| $\underset{\sigma}{Sim}(G^a, G^u)$ | the similarity between the anonymized graph and auxiliary graph under the mapping $\sigma$ |
| $c_i, c_j$ | the closeness centrality of node $i$ in anonymized graph and node $j$ in auxiliary graph |
| $S_C(i,j)$ | the correlation similarity between node $i$ and $j$ |

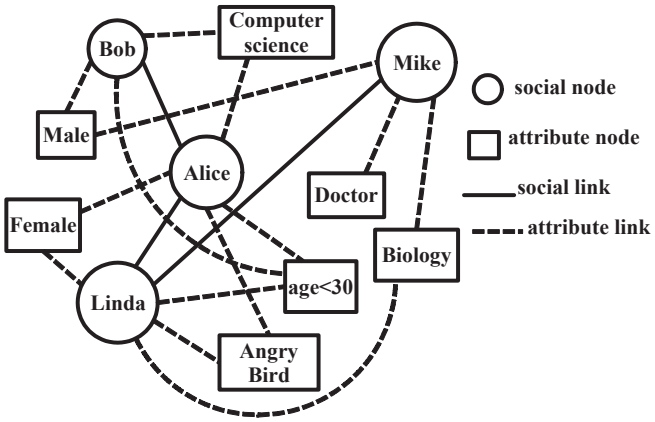



Fig. 1. An example of Social-Attribute network

**Auxiliary graph.** The auxiliary graph is modeled by $G^u = (V^u, E^u, t^u)$ where $V^u$ is the set of nodes, and $E^u$ represents all links between the nodes, that is $E^u = \{e_{i,j}^u | i, j \in V^u, a\ social\ tie\ exists\ between\ i\ and\ j\}$, and $m^u = |E^u|$ represents the number of edges. Then, $t^u$ represents the node type, for instance, $t_j^u$ represents the type of node $j$, when $t_j^u = S$, it means that node $j$ is a social node, and when $t_j^u = A$, it means node $j$ is an attribute node. Given $\forall j \in V^u$, its neighborhood is defined as $N^u(j) = \{i \in V^u | e_{i,j}^u \in E^u\}$. Then we define $\Delta_j^u = |N^u(j)|$ as the number of neighbor nodes of $j$. The attribute information of the node $j$ can be expressed as a $d$-dimension binary vector $\overrightarrow{b_j^u}$, thus the matrix $B^u$ representing all attribute values of all social nodes.

### Attack Model

The goal of de-anonymization is to map the nodes in $G^a$ to the nodes in $G^u$ as accurately as possible. In a real social network, the auxiliary network can be obtained in a variety of ways, such as data mining, cooperative information system, knowledge/data attacks. Given $G^a$ and $G^u$, we can employ a mapping to formally define a de-anonymization attack: $\sigma : V^a \rightarrow V^u$. For $\forall i \in V^a$, its mapping under $\sigma$ is $\sigma(i) \in V^u \bigcup \{\perp\}$, where $\perp$ is a special not existing indicator. Under $\sigma$, a successful de-anonymization attack on $i \in V^a$ is defined as $\sigma(i) = i'$, if $i' \in V^u$ and $i$ and $i'$ correspond to the same user; or $\sigma(i) = \perp$. Otherwise, other cases imply that the attack on $i$ fails. Accordingly, our goal of a de-anonymizaiton attack is to successfully de-anonymize as many users in $V^a$ as possible.

## 4 Spectral Graph Partition

Social networks in real world can be very large. So the computation cost of different structural characteristics such as clustering coefficient are expensive. Therefore, we adopt the spectral graph partition algorithm in [36] to partition the large social network into a number of small graphs and then de-anonymize them in parallel. From another aspect, graph partition can effectively reduce the error accumulation of de-anonymization. We just introduce the algorithm briefly, which can be referred to the citation for details. For the graph $G = (V, E)$, let matrix $A = |a_{i,j}|$ be the weighted adjacency matrix of graph $G$. Notice that in this paper, graph $G$ is unweighted graph, the edge weights can be considered to be all one. So the matrix $A$ of graph $G = (V, E)$ be defined through it elements:

$$a_{i,j} = \begin{cases} 1 & if(i,j) \in E \\ 0 & otherwise \end{cases} \quad (1)$$

And $D$ is a diagonal matrix that $D = diag(Ae)$, where vector $e = (1, \ldots, 1)^T$. Actually, for the unweighted graph $G$, the entry $a_{i,j}$ denotes an edge between node $i$ and $j$, and the diagonal matrix $D$ denotes the degree of each node. The Laplacian matrix $L$ is defined as $L = D - A$.

Let a set of vertices $S \subseteq V$ then its boundary is a set of edges $\partial(S) \subset E$ that only one end point vertex of each edge is in $S$, in other words,

$$\partial(S) = \{(i,j) : i \in S \bigwedge j \notin S\} \tag{2}$$

and let a cut $C = (S, T)$ of a graph be a partition of vertices V into two disjoint sets $S, T \subseteq V$. For the problem of graph partitioning, our goal is to find a minimum balanced cut $C = (S, \bar{S})$ of a graph $G = (V, E)$ in the sense that $S$ satisfied either the condition

$$\rho(S) = \min_S \frac{vol(\partial(S))}{|S||\bar{S}|} \tag{3}$$

or

$$\eta(S) = \min_S \frac{vol(\partial(S))}{|vol(S)||vol(\bar{S})|} \tag{4}$$

where $\bar{S}$ is the complement set of $S$ with respect to $V$, $|\cdot|$ denotes the cardinality of a set. The cost function $\rho(S)$ is often referred to as ratio cut, while the $\eta(S)$ is referred to as normalized cut.

$$vol(\partial(S)) = |\partial(S)| \tag{5}$$

$$vol(S) = \sum_{i \in S} d(i) \tag{6}$$

The graph partitioning in our solution is shown in Algorithm 1. [36]

**Algorithm 1:** Spectral Graph Partitioning

1: Let $G = (V, E)$ be an input graph and $A$ be its adjacent matrix, $D$ be the degree matrix.
2: Let $B = I$ is the ratio cut.
3: Let $p$ be the number of desired partitions, then compute the Laplacian matrix $L = D - A$
4: Find $p$ smallest eigenpairs of the eigenvalue problem $LU = BU\Sigma$, of which $\Sigma = diag([\lambda_1, \ldots, \lambda_p])$
5: Scale the $p$ eigenvectors $U$ by row or by column.
6: Run $k$-means clustering algorithm on points defined by rows of $U$, so that the similar vertices could be clustered together.

Consequently, the large social graph can be divided into several small subgraphs. Both the anonymized graph and the auxiliary graph are divided into subgraphs by the same approach. Then corresponding subgraphs can be matched according to the $k$ largest eigenvalues. Finally, all the subgraphs of the two social graphs can be matched one by one. Moreover, the de-anonymization of nodes in the subgraphs can be executed according to structural characteristics and attribute information to achieve the identification of social network users.

## 5 De-anonymization

### Similarity Measurement

Firstly, we formally define the similarity measurement of nodes with the consideration of structure characteristics and attributes of social networks before introducing the de-anonymization algorithm. The similarity $S(i,j)$ between nodes $i, j$, includes (obviously, here we only measure the similarity between social nodes), the attribute similarity and the structural similarity.

**Attribute Similarity.** The attribute similarity of node $i \in V^a, t_i^a = S$ and $j \in V^u, t_j^u = S$ is denoted as $S_A(i,j)$. For each node $i \in V^a, t_i^a = S$ in $G^a$, the vector $\vec{b_i^a}$ represents the attribute values of node $i$. Similarly, for each node $j \in V^u, t_j^u = S$ in $G^u$, the vector $\vec{b_j^u}$ represents the attribute values of node $j$. So the vector $\vec{b_i^a}$ and $\vec{b_j^u}$ can be shown as:

$$\vec{b_i^a} = (b_{1i}^a, b_{2i}^a, b_{3i}^a \cdots b_{di}^a)$$

$$\vec{b_j^u} = (b_{1j}^u, b_{2j}^u, b_{3j}^u \cdots b_{dj}^u)$$

of which $d$ is the dimension of the vector, that is, the $d$ is the number of all attribute values.

$$S_A(i,j) = \frac{\vec{b_i^a} \bullet \vec{b_j^u}}{\sum_{x=1}^{d} (b_{xi}^a \bigoplus b_{xj}^u) + \vec{b_i^a} \bullet \vec{b_j^u}} \tag{7}$$

From the definition of $S_A(i,j)$, we can see that it is between 0 and 1. If the two nodes have more of the same attribute values, then their attribute similarity is high.

**Structural Similarity.** In addition to the attribute similarity of nodes, we also consider the structural similarity between nodes. We adopt $S_R(i,j)$ to represent the structural similarity between node $i \in V^a, t_i^a = S$ and node $j \in V^u, t_j^u = S$. The number of neighbors of node $i$ in $G^a$ is denoted as $\Delta_i^a = |N^a(i)|$. We compute the number of social neighbors of node $i$, denoted as $(\Delta_i^a)^S = |\{v \in N^a(i) | t_v^a = S\}|$. Moreover, we develop the 1-hop degree-based model.The detailed procedure can be shown in Algorithm 2. Here we use to $N(i,1)$ to emphasize the one-hop neighbor node. Therefore, the structural similarity between two nodes is:

$$\begin{aligned} S_R(i,j) &= 1/2(1 - \frac{|(\Delta_i^a)^S - (\Delta_j^u)^S|}{max\{(\Delta_i^a)^S, (\Delta_j^u)^S\}}) \\ &+1/2(\frac{count}{max\{|N(i,1)|, |N(j,1)|\}} \times \frac{\overrightarrow{A_i'} \bullet \overrightarrow{B_j'}}{\|\overrightarrow{A_i'}\| \times \|\overrightarrow{B_j'}\|}) \end{aligned} \tag{8}$$

**Node similarity.** The node similarity between node $i \in V^a, t_i^a = S$ and node $j \in V^u, t_j^u = S$ is denoted as $S(i,j)$, then it is computed as

$$S(i,j) = 1/2 S_A(i,j) + 1/2 S_R(i,j) \tag{9}$$

As mentioned in our attack model, a de-anonymization scheme can be defined as a mapping: $\sigma = V^a \rightarrow V^u$. Therefore, our goal of the de-anonymizaiton is to find a mapping $\sigma$ which maximizes the similarity between $G^a$ and $G^u$. We use the function $Sim$ to measure the similarity between $G^a$ and $G^u$ after nodes matching by $\sigma$.

**Algorithm 2:** 1-hop degree-based model

**for** each $i \in V^a$ and $t_i^a = S$ **do**
  Define $N(i,1)$, which represents the 1-hop social neighbor nodes of node $i$, where $x_1, x_2 \cdots \in N(i,1)$. (Here we use to $N(i,1)$ to emphasize the one-hop neighbor)
  Sort the degree sequence in descending order, $d(x_1) \geq d(x_2) \cdots$, of which $d(x_i)$ represents the degree of node $x_i$
  (We use $d(x_1)$ to represent the degree for simplicity)
**end for**
**for** each $j \in V^u$ and $\mathrm{t_j^u} = \mathrm{S}$ **do**
  Define $N(j,1)$, which represents the 1-hop social neighbor nodes of node $j$, where $y_1, y_2 \cdots \in N(j,1)$
  Sort the degree sequence in descending order, $d(y_1) \geq d(y_2) \cdots$, of which $d(y_i)$ represents the degree of node $y_i$
**end for**
$\overrightarrow{A_i} = (d(x_1), d(x_2), \cdots d(x_{|N(i,1)|}))$
$\overrightarrow{B_j} = (d(y_1), d(y_2), \cdots d(y_{|N(j,1)|}))$
**while** compute $S_R(i,j)$ **do**
  **for** each vector element of $\overrightarrow{A_i}$ **do**
    Compute $|d(x_i) - d(y_j)|$ with all the vector elements of $\overrightarrow{B_j}$
    Record the $(d(x_i), d(y_j))$ as the matching pair which satisfies that $|d(x_i) - d(y_j)|$ is minimized
  **end for**
  Compare all $|N(i,1)|$ matching pairs
  **if** more than two elements of $\overrightarrow{A_i}$ are matched to the same element of $\overrightarrow{B_j}$ **then**
    Compare the values of these matching pairs, delete the matching pair with bigger value
  **end if**
  Else
  Record all the matching pairs $(d(x_i), d(y_j))$, and denote the number of matching pairs as $count$
  Let each vector element of $\overrightarrow{A_i'}$ equals the left value of each matching pair
  Likewise, let each vector element of $\overrightarrow{B_j'}$ equals of the right value of each matching pair
**end while**

$$\underset{\sigma}{Sim}(G^a, G^u) = \sum_{(i,\sigma(i)=j)\in\sigma} S(i,j) \tag{10}$$

Consequently, the de-anonymization problem can be defined as follows:

**Definition 1.1 De-anonymization Problem**

**Input:** The anonymized graph $G^a$ and auxiliary graph $G^u$

**output:** A best map $\sigma$

**Goal:** To maximize the similarity between $G^a$ and $G^u$, $\arg\max \underset{\sigma}{Sim}(G^a, G^u)$.

**De-anonymization Algorithm**

The objective of our de-anonymization attack is to find a map which maximizes the similarity of social nodes in $G^u$ and $G^a$. To transform this problem, we construct a complete weighted bipartite graph $G^B = (V^a + V^u, \varepsilon^B)$, where a weight $S(i,j)$ is assigned to each link $e_{ij} \in \varepsilon^B$. That is, we connect all links between all social nodes. Then the de-anonymization problem can be reduced to the maximum weighted bipartite matching problem, thus can be solved by the Hungary algorithm and KM algorithm. However, the social networks in real world can be very large. Therefore, constructing the complete bipartite graph has large time and space complexity. Furthermore, finding the maximum weighted matching also is a great challenge for large social networks.

So a light-weighted bipartite graph is built in Algorithm 3. We transverse the social nodes in $G^a$ and calculate the similarity with the social nodes in $G^u$ to find the first $k$ nodes with the highest similarity, of which $k$ is the predefined parameter. Finally, a mapping is obtained that each node corresponds to $k$ matching nodes. In the beginning, $G^B$ has no links. The breadth first search (BFS) is performed on $G^a$. An outstanding initial node is selected from $G^a$ so that it can be mapped with high accuracy. This initial node can be selected by many different ways. In our algorithm, we randomly selected a node and calculate the similarities between it and all nodes in $G^u$ to examine if there is a successful mapping (We match the node in $G^a$ with the corresponding node in $G^u$ who has the largest similarity, which is greater than a threshold). Then the node can be selected as the initial node.

Then the BFS is performed in $G^a$. In this process, our strategy is that if two nodes match, then their neighbors are likely to match. Before mapping each social node $i \in G^a$, the algorithm first checks whether $i$ has a predecessor node, which is denoted as $pr(i)$. If so, it searches the $pr(i)$'s candidate nodes and their neighbors, computes the similarity and then compares the similarity with the threshold $r$. If the similarity is less than $r$, then we delete this candidate node. Otherwise, it searches all the nodes in $V^u$.

**Line 3-19 in Algorithm 3:** For each node $i \in V^a \; and \; t_i^a = S$, we firstly verify that the node $i$ has a predecessor node $pr(i)$. If there is a predecessor node, we can search the neighbors of $pr(i)$'s candidate nodes to get the top $k$ similar candidate nodes of the node $i$ in this range.

**Line 21-27 in Algorithm 3:** If node $i$ does not have a predecessor node, each $j \in V^u \; and \; t_j^u = S$ is traversed directly to find the first $k$ candidate matching nodes.

**Line 28-34 in Algorithm 3:** Therefore, $k$ links from $i$ to the candidate nodes are added to the graph $G^B$, and the similarity score between the node $i$ and its candidate node is the weight of the edge. So the bipartite graph is obtained. The classical KM algorithm is adopted to find a maximum weighted bipartite matching based on this bipartite graph.

In our algorithm, only the parent node needs to compare with all the nodes to obtain the first $k$ nodes with the highest similarity. The most nodes only compare with the neighbors of his predecessor node's candidate matching nodes to obtain the first $k$ nodes with the highest similarity, which greatly decreases the time complexity. Moreover, if

we construct a complete bipartite graph, we need to transverse all the nodes in the graph. The number of links in this complete bipartite graph is $O(n_s^a \cdot n_s^u)$(the number of social nodes in $G^a$ and $G^u$). To reduce the mapping complexity, we can decrease the links by keeping only links with the top $k$ largest weights. Each node in $V^a$ is linked to top $k$ candidate nodes in $V^u$. Accordingly, the number of links is reduced to $O(kn_s^a)$. Thus the time and space complexity of solving the maximum weighted bipartite matching problem is lowered respectively.

Herein, $k$ and $r$ are the predefined parameters to balance the accuracy of the de-anonoymization and the complexity of the algorithm. So $k$ and $r$ affect the accuracy of the final mapping results. When the value of $k$ is too large and $r$ is too small, it will lead to a bipartite graph with many unnecessary edges for the mapping; if $k$ is too small and $r$ is too large, it will make the matching process missing some important links. Therefore, we will evaluate the impact of the parameters through experiments.

## 6 SIMILARITY ANALYSIS OF DIFFERENT ANONYMIZED METHODS

### Similarity Analysis

The structural characteristics of social networks can be the reference for de-anonymization, while both the anonymized and auxiliary data can be modeled as graphs. The de-anonymization based on structural characteristics is meaningful because we also adopt other approaches to refine the coarse granularity de-anonymiztion. In this section, we present three widely accepted measurements to discuss the topological properties of social nodes, namely degree distribution, closeness centrality and correlation similarity respectively[32]. In particular, it is needed to be explained that we only consider the social nodes and social links of the data graph in this section.

**Degree Distribution.** The degree distribution considers the number of edges that a node has (or the number of neighbor nodes) in a graph. In the anonymized data graph, the degree of $i \in V^a$ is defined as $\Delta_i^a = |N^a(i)|$, similarly $\Delta_j^u = |N^u(j)|$ for $j \in V^u$ which are defined in Section 3.

**Closeness Centrality.** The degree distribution indicates the local property of nodes in social network since we only consider the adjacent links. To fully characterize the topological property of nodes, it is useful and significant to describe the measurement of closeness centrality defined from a broader and global view. The closeness centrality measures how close a node is to all other nodes in a graph, which is defined as the ratio between $n-t-1$ and the sum of its distances to all other nodes (if the number of social nodes is $n$), of which $t$ is the number of nodes which can't be reached. Formally, the closeness centrality $c_i$ for node $i \in V^a$ is shown as follows:

$$c_i = \frac{|V^a| - t - 1}{\sum\limits_{x \in V^a, x \neq i} |p^a(i,x)|} \tag{11}$$

of which $p^a(i,x)$ represents the shortest path from node $i$ to node $x$, and $|p^a(i,x)|$ represents the hops of this path.

**Algorithm 3:** SA-based De-Anonymization (SA-DA)

**Input:** Anonymized graph $G^a = (V^a, E^a, t^a)$;
Auxiliary graph $G^u = (V^u, E^u, t^u)$ , parameter $k$, $r$.
**Output:** a maximum weighted bipartite matching $\sigma$

Define $\varepsilon^B = \emptyset$, build a bipartite graph $G^B = (V^a + V^u, \varepsilon^B)$;
Define $Can = \emptyset$, which represents the candidate matching nodes
Perform BFS in $G^a$ starting from $p_0$(an initial node $p_0 \in G^a$)
**for** each $i \in V^a$ and $t_i^a = S$ following the BFS order **do**
  **if** $i$ has a predecessor $pr(i)$ **then**
    Define $N = \emptyset$
    **for** each $v \in Can_{pr(i)}$ **do**
      **for** each neighbor $j$ of $v$ **do**
        $N = N \bigcup \{j\} \bigcup \{v\}$
      **end for**
    **end for**
    **for** each $n \in N$ **do**
      Compute $S(i,n)$
      **if** $S(i,n) > r$ **then**
        $Can_i = Can_i \bigcup \{n\}$
      **end if**
    **end for**
    Select the top $k$ similar nodes in $Can_i$ as $i$'s candidates
  **end if**
  Else
  **for** each $j \in V^u$ and $t_j^u = S$ **do**
    Calculate $S(i,j)$
    **if** $S(i,j) > r$ **then**
      $Can_i = Can_i \bigcup \{j\}$
    **end if**
  **end for**
  Select the top $k$ similar nodes in $Can_i$
  **for** each $a \in Can_i$ **do**
    Attach the node $i$, $a$, and $\varepsilon^B = \varepsilon^B \bigcup e_{i,a}$
  **end for**
**end for**
Construct the bipartite graph $G^B$
Execute the KM algorithm on the bipartite graph $G^B$
Return a maximum weighted bipartite matching $\sigma$

Similarly, the closeness centrality $c_j$ of $j \in V^u$ is defined as:

$$c_j = \frac{|V^u| - t - 1}{\sum\limits_{x \in V^u, x \neq j} |p^u(j,x)|} \tag{12}$$

**Correlation Similarity.** It is observed that two nodes are likely to match if they have more common neighbors that have been mapped. Here, we denote correlation similarity as $s_C(i,j)$ which can be shown in equation 13 for node $i \in V^a$ and $j \in V^u$. First, $M(i,j)$ represents the neighbors which have been mapped of node $i$ and $j$. $M(i,j) = \{(x,x') | x \in N^a(i), x' \in N^u(j), (x,x') \in \sigma\}$.

$$S_C(i,j) = (1 - \frac{|(\Delta_i^a)^S - (\Delta_j^u)^S|}{max\{(\Delta_i^a)^S, (\Delta_j^u)^S\}}) \cdot \frac{\sum_{(x,x') \in M(i,j)} S_R(x,x')}{|M(i,j)|} \quad (13)$$

of which, $S_R(x,x')$ is the structural similarity between $x$ and $x'$ which is formally defined in section 5.

We have defined three measurements to evaluate the network properties of the anonymized graphs and the original one as well as how close they are based on different anonymization methods. Therefore, to anonymize the social networks for different datasets, we employ 3 popular anonymization techniques, namely Ran Add/Del, partitioning and summarizing Anonymity [9] and Union-Split Anonymity(UniSpl) [37] with their default parameters. We briefly describe these anonymization techniques, which can be referred to the citations for elaborate details.

- **Rand Add/Del**, we randomly insert one edge followed by deleting another edge and repeat this process for many times. This strategy keeps the total number of edges unchanged in the original graph. Here we randomly delete $4\%$ links of the social network and randomly insert the same number of edges.
- **Partitioning and summarizing** [9], this algorithm anonymizes a graph by grouping nodes into partitions, and describes the graph at the level of partitions. Then the number of nodes in each partition will be published, along with the density of edges that exist within and across partitions. Therefore, the output of this algorithm is a generalized graph.
- **UniSpl** [37], this algorithm clusters individuals into groups in social networks with similar social roles, while satisfying a minimum cluster size constraint. Then edges are added and removed strategically based on the node's inter-cluster connectivity.

### Experimental Analysis

Now we discuss how graph anonymization techniques affect network properties. The network properties explained above are commonly measured and reported on the realistic dataset of Twitter. After analyzing the data we crawled from the Twitter dataset, the social network includes $7910$ social nodes, $874222$ social links. We consider the degree distribution of all nodes, then we randomly select some nodes to record and compare their different network properties.

As shown in Fig.2(a), it is observed that the degree of Twitter dataset follows the power-law distribution. Moreover, the degree distribution of the anonymized graph and the original graph are similar for the algorithm of Rand Add/Del. For the algorithm of partitioning and summarizing anonymity, the degree distribution of the anonymized graph are qualitatively similar to the original graphs, which is shown in Fig.2(b). Besides, the high degrees are reduced systematically by the graph processing. Similarly, the graph anonymized by the method of UniSpl are similar to the original graphs of the degree distribution as demonstrated by Fig.2(c). This method performs well in preserving network properties by attempting to retain local network structure and global structure. In the simulation process, we set $k = 8$ of this algorithm. It is worthwhile to note that while $k$ is relatively large(such as $k = 50$), the anonymized graph has a larger deviation from the original one in virtue of the degree requirement of $k$-anonymity.

Therefore, we can conclude that degree distribution can be used for de-anonymization. Moreover, multiple nodes in the anonymized and original graphs also have similar degree as shown in Fig.2(d) (we randomly select twenty nodes of the Twitter dataset), which also illustrates that degree distribution can be used for de-anonymization.

Then, since the number of nodes in dataset is too big, we also randomly selected 20 nodes in the Twitter dataset to compute and compare their closeness centrality. As shown in Fig.2(e), it presents closeness centrality of the randomly selected nodes in the anonymized graph and the real mapping nodes in the original graph based on the three anonymized methods. Also the closeness centrality of nodes in the anonymized graph agrees with that in the original graph to a certain extent. Therefore, it suggests that the closeness centrality can be a measure of utility to evaluate how close the anonymized graph to the original one.

Since some nodes with distinguished structural property, they agree with their real mapping nodes and significantly disagree with other nodes in the original graph. Consequently, it indicates that even just based on some structural property, these nodes with distinguished characteristics can be de-anonymized successfully. However, for the nodes with indistinctive structural similarities, it is difficult and impossible to achieve the exact mapping just based on structural property especially for different anonymization techniques. So we need to consider other proper and effective multimeasurements collaboratively.

Based on the definition of correlation similarity $S_C(i,j)$, we can find that two nodes are likely to match if they have more common neighbors which have been mapped when executing the de-anonymization, and resulting in high correlation similarity score. We also consider the effect of degree difference between two nodes. When it is small, the correlation similarity is greater. Based on the three anonymization methods, it is assumed that half of the nodes have been mapped. Then we randomly selected $13$ nodes from the rest of nodes, their correlation similarity between the anonymized graph and the auxiliary graph is shown in Fig.2(f), 2(g), 2(h) for the Twitter dataset. In this figure, real mapping means that the correlation similarity between $v \in V^a$ and its successful node mapping. Max represents $max\{s_C(v,x)|x \in V^u, x \neq \sigma(v)\}$. Min shows $min\{s_C(v,x)|x \in V^u, x \neq \sigma(v)\}$ and average denotes $\frac{1}{|V^u-1|}\sum_{x \in V^u, x \neq \sigma(v)} s_C(v,x)$. It is important to note that the nodes we selected are different in three anonymization methods. Under the assumption that half number of nodes have been mapped, some nodes such as nodes 3, 6, 11 in Fig.2(f), nodes 2, 6, 10 in Fig.2(g), nodes 4, 12, 13 in Fig.2(h) agree with their real mapping nodes while significantly disagreeing with all other nodes in the auxiliary graph. It implies that we can take the correlation similarity into consideration because sometimes nodes are potentially easier to be de-anonymized under this metric. However, it doesn't perform well for all the nodes.

In summary, the differentiability of anonymized nodes is distinct based on diverse anonymized techniques as well

as different similarity measurements. The analysis on the datasets suggests us to define a multimeasurement with the consideration of multiple similarity metrics and measurement for effective de-anonymization.

## 7 Data collection

In this section, we study and discuss the performance of our presented de-anonymization algorithm on three real datasets. First, we collect a dataset from Twitter. Specifically, we collected social structures (including the social users and the relationship between them) and user attributes from Twitter by using Twitter APIs. We first collected a social network with user attributes by iteratively crawling the social users and their friends. Subsequently we crawled the user id, screen name, user name, create time, city, time zone and biography. Due to the Twitter API limits, our program only access the most recent 3000 tweets and the most recent 7000 friend ids for each social user. Secondly, the dataset of Facebook is obtained from Stanford Network Analysis Projects(SNAP) [39]. It is a popular online social network which contains rich network data and users' file. We also collected the social nodes and social links, with some user attributes. However, some of user attributes are anonymized. Then, we also execute our algorithm on the arXiv dataset [40] to evaluate its performance. The dataset is crawled from the arXiv Condensed Matter E-print Archive, which contains all the e-prints of scientific papers in the category of cs.NI(Network and Internet Architecture) and cs.CR(Cryptography and Security). All the collected information is publicly available.

### Twitter DataSet

Each user in Twitter has the unique id, and friend lists. The Twitter website provides the API to share their data. So we can crawl the data using the Twitter API. We construct an undirected social network by keeping an undirected link between user $a$ and user $b$ if $a$ is in $b$'s friend list or $b$ is in $a$'s friend list. Our dataset consists of $7910$ users and $874222$ undirected social links after the processing.

**User attributes:** The data we crawled from Twitter includes the attributes of user id, gender, screen name, user name, create time, city, time zone and biography. We consider two attributes of create time and cities. We get the content of Twitter profile as users' attributes. It is noted that users fill in their profiles freely resulting in many infrequent attribute values or meaningless and duplicate attribute values. Moreover, small typos of inputs sometimes make the same attribute value be different. So when we preprocess it, the incomplete or invalid or duplicate attributes are removed. Therefore, we label the attribute values specifically.

(1) Create time. The Twitter website will record the "create time" of each user. We adopt the time as one of user's attribute. The time is specific to the minute. When it is used as the user attribute, we just quote the "year" as the attribute value. So we consider nine attribute values as $2005 - 2013$.

(2) Cities. We select the top-70 cities in which most users claimed they have lived in. Afterwards, we process the crawling data by manually merging cities which actually refer to the same one, then 70 distinct cities are obtained.

TABLE 2
Basic statics of the SA network

| Dataset | Social nodes | Social links | Attribute nodes | Attribute links |
|---|---|---|---|---|
| Twitter | 7910 | 874222 | 79 | $7910 * 2$ |
| Facebook | 4032 | 88234 | 112 | $4032 * 3$ |
| arXiv | 17955 | 34976 | 92 | $17955 * 3$ |

In total, we consider 79 distinct attribute values, including 9 time and 70 cities. It is acknowledged that our dataset might not be a representative sample of the recent entire Twitter network.

### Facebook DataSet and arXiv

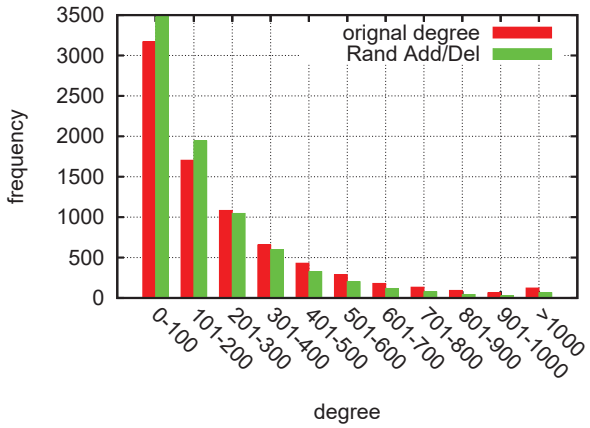
For Facebook dataset, it contains $4032$ nodes and $88234$ undirected social links. The data we crawled contains the education school, hometown and some anonymized features. So we do some process for the network data such as removing the invalid and duplicate attributes. Moreover, we add one user attribute of gender for the social network by randomly assigning the attribute value to the social network users. So after processing, it contains three attribute values of "education school", "hometown" and "gender". In total, $112$ distinct attribute values are considered including $40$ education schools, 70 hometowns and 2 gender attributes.

For the dataset of arXiv, the authors are considered as the vertices in the social network. If the scientists coauthor the same paper, there exist social links between them. It contains $17955$ nodes and $34976$ social links in the constructed social networks. The website records the "time" of each paper. So we adopt the time as one of users' attribute. The time is specific to the minute. When it is used as the user attribute, we just quote the "year" as the attribute value. So we consider $20$ attribute values as $1996 - 2015$. Then we also execute some process on the dataset. We add two attributes to each user including school of author and gender. We select the top-70 schools of USA and randomly assign to each user. Similarly, the attribute value of "gender" is assigned to each user in the same way. In total, $92$ distinct attribute values are considered including $70$ schools, $20$ time and $2$ gender attributes.

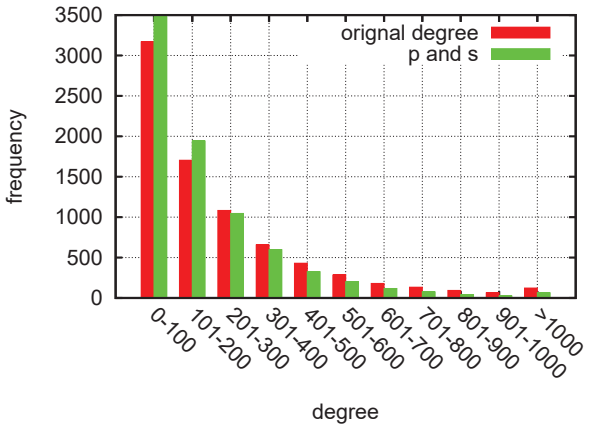
### Constructing SA Frame work

We take each user as a social node and the friendship between them as the social links of three datasets. For each user, we consider the attribute values thus adding many attribute nodes. If a user has the attribute value, we create a link between the social node and the corresponding attribute node. Table 2 shows the basic statistics of our constructed SA framework.

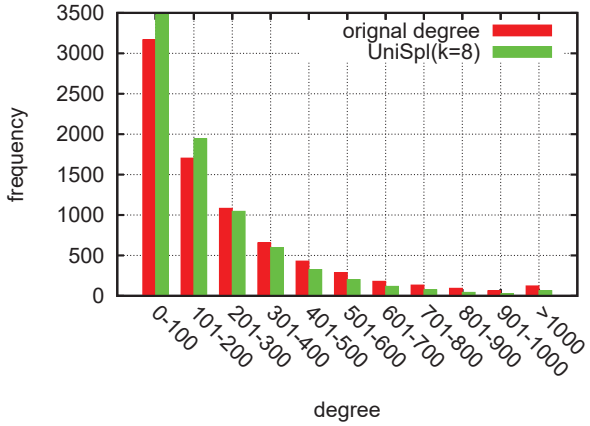
**Anonymized Graph:** We firstly need to generate an anonymized graph before conducting the de-anonymization algorithm on datasets. Both the nodes and the links should be perturbed to obtain an anonymized graph. In this paper, we use the first anonymization method mentioned in section 6. We will execute some perturbation on the network graph. First, the social nodes are perturbed by removing the node identity and substituting with the random generated



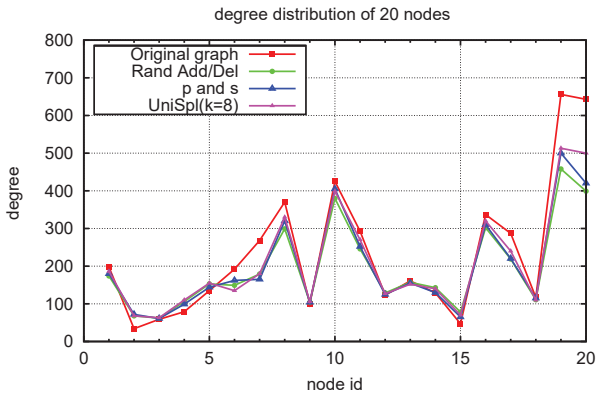

(a) degree distribution of Ran Add/Del



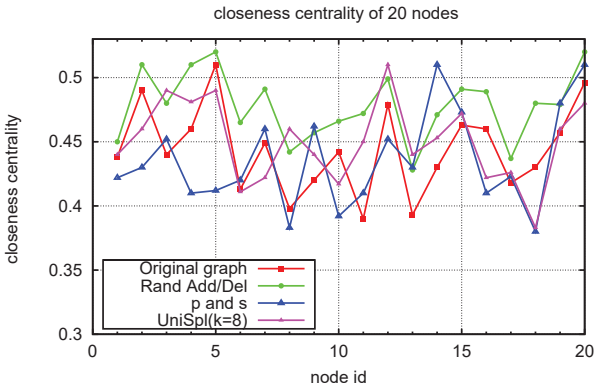

(b) degree distribution of partitioning and summarizing



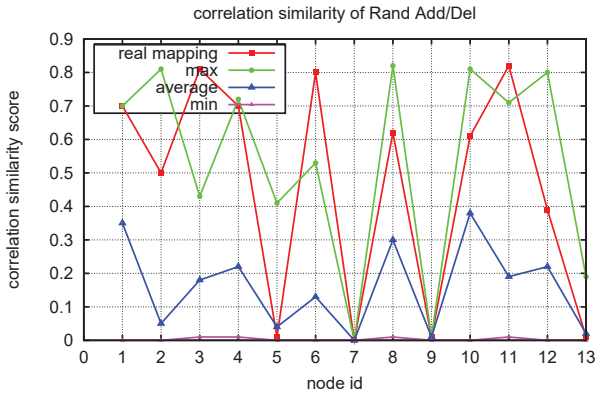

(c) degree distribution of UniSpl



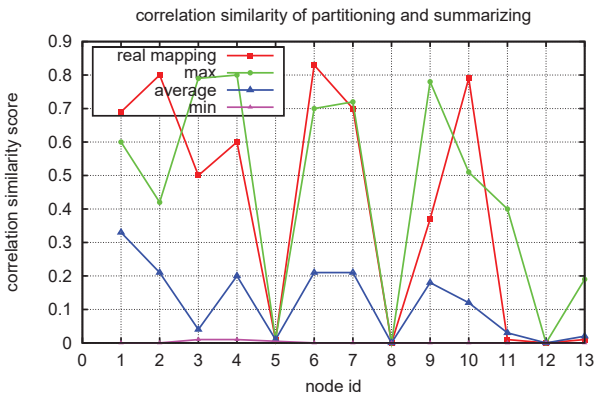

(d) node degree of 20 nodes



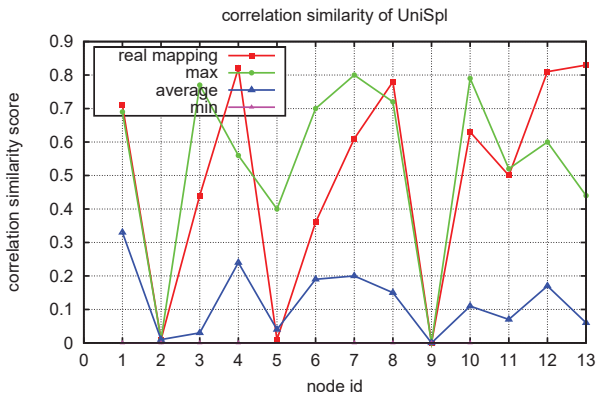

(e) average distance centrality of 20 nodes




(f) correlation similarity of Rand Add/Del




(g) correlation similarity of partitioning and summarizing




(h) correlation similarity of UniSpl

Fig. 2. The evaluation analysis of Twitter dataset.

anonymous identifiers. Then we randomly delete some links and then randomly insert the same number of edges. In our experiment, if we want to add $p$ noise to the anonymized graph, we randomly delete $(p/2) \times |E^a|$ links and then add the same number of links.

## 8 Experiment Evaluations

We describe metrics of accuracy (the ratio of successful matchings) and run time to evaluate our de-anonymization algorithm, which are used to measure the utility and time complexity. The impact of different parameters are also evaluated in this section. Our presented de-anonymization algorithm is exploited on three datasets of Twitter, Facebook and arXiv.

For the Twitter dataset, Fig.3 shows the impact of the parameters of $k$ and $r$ on the accuracy and run time. As depicted in Fig.3(a), the success rate can be improved with the increasing $k$ when $k \leq 8$. However, when $k > 8$, the improvement is not obvious. Moreover, the run time increases with the growing $k$ constantly as shown in Fig.3(b). So we often choose $k = 8$ as default. Then as depicted in Fig.3(c) and Fig.3(d), when $r$ increases, the obtained bipartite graph is reduced and some real links will be missed, which may cause the node matching to fail, so the success rate and run time greatly decrease. Similarly, for the dataset of Facebook and arXiv, the simulation results in Fig.4 and Fig.5 also demonstrate the conclusion.

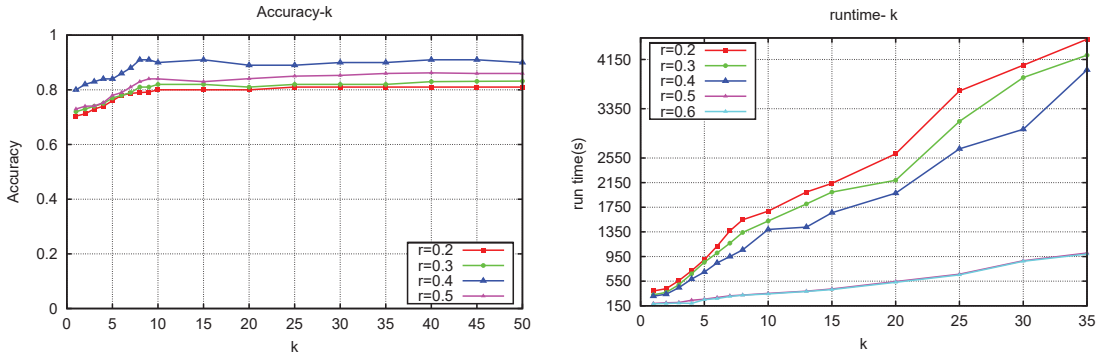
In Fig.6(a), we discuss the impact of perturbation of the anonymized graph on the matching accuracy. For the twitter dataset, the accuracy rate does not decrease too much while the perturbation rate increases, which varies from $0.91$ to $0.84$. But for the dataset of Facebook and arXiv, the perturbation rate has an obvious effect on the accuracy. It has a great drop from $0.903$ to $0.72$ for Facebook dataset, from $0.803$ to $0.64$ for arXiv dataset. That's because the number of social nodes is not too much and the number of social links is very large in Twitter dataset resulting in the more average degree of the social network. So when we execute the perturbation on the social graph, the change is not so obvious. Consequently, it suggests that sometimes the de-anonymization results may be different based on different datasets, that is, instance-specific.

Fig.6(b), 6(c), 6(d) show the de-anonymization accuracy of different methods on the three datasets, including the methods of RF [41], ADA [32], RFC [42] and our algorithm. It can be seen that our method achieves the best performance. Even for high perturbation, our method can still guarantee high accuracy. It is worth noting that when more than $20\%$ links are changed, the structure of social network graph is significantly changed. For data integrity, the data publisher will not change dramatically in real datasets.

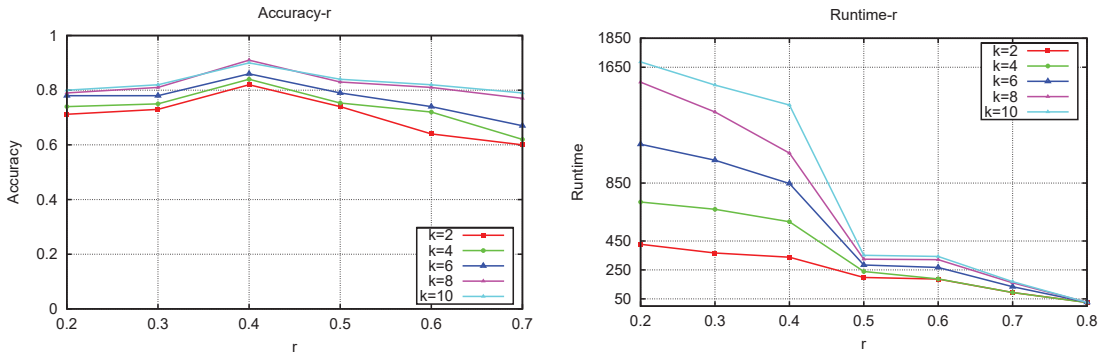
**Discussion** The good performance of our algorithm mainly can be attributed to the consideration of both the attribute similarity and structure similarity. It can greatly improves the matching accuracy. In our evaluation, we only use two or three attributes. The more attributes we adopted, the more the accuracy will be. On the other hand, the time and space complexity correspondingly increase much when we adopt the attribute matrix. But if we consider it carefully, the matrix is a sparse matrix. So when we analyze the matrix and calculate the node similarity, we can perform some corresponding compression processing and calculation skills.

## 9 Conclusion

In this paper, we construct an SA framework network which models the social network. The social network users, attributes, and behaviors (in this model, the user behavior is treated as an attribute) are integrated. Based on the SA network model, the de-anonymization attack on the social network is conducted. The method proposed in our paper also considers node attribute similarity and structural similarity for node matching, thus improving the accuracy of de-anonymization. Through three realistic social network datasets, the accuracy and efficiency of the de-anonymization method are verified. Moreover, we discuss how graph anonymization techniques affect network properties and provide similarity analysis based on different anonymized methods. The elaborate analysis suggests us to define a multimeasurement with the consideration of multiple similarity metrics for effective de-anonymization.

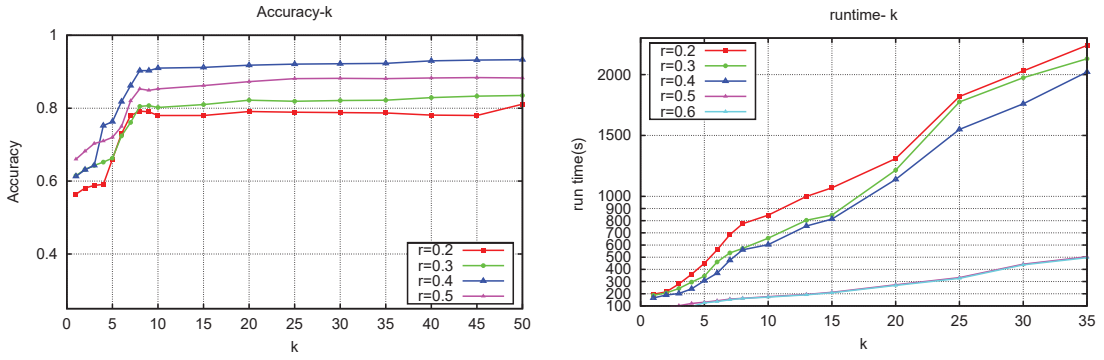



(a) Accuracy-$k$ (b) Runtime-$k$

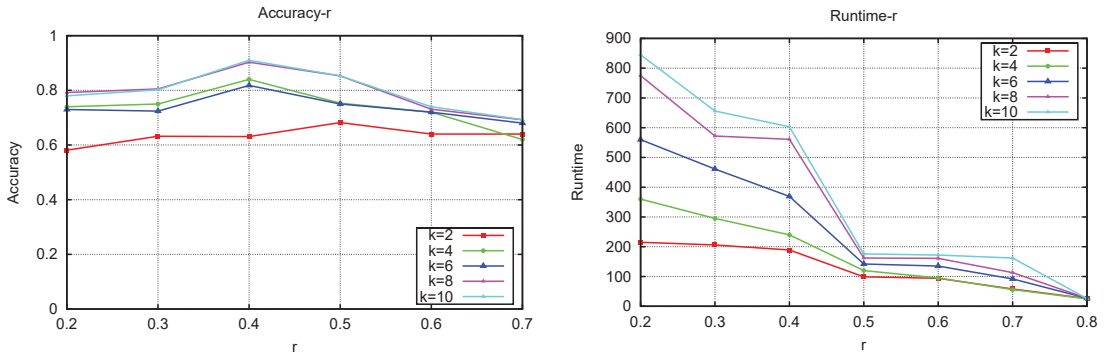



(c) Accuracy-$r$ (d) Runtime-$r$

Fig. 3. Experimental results on Twitter dataset with $4\%$ noise

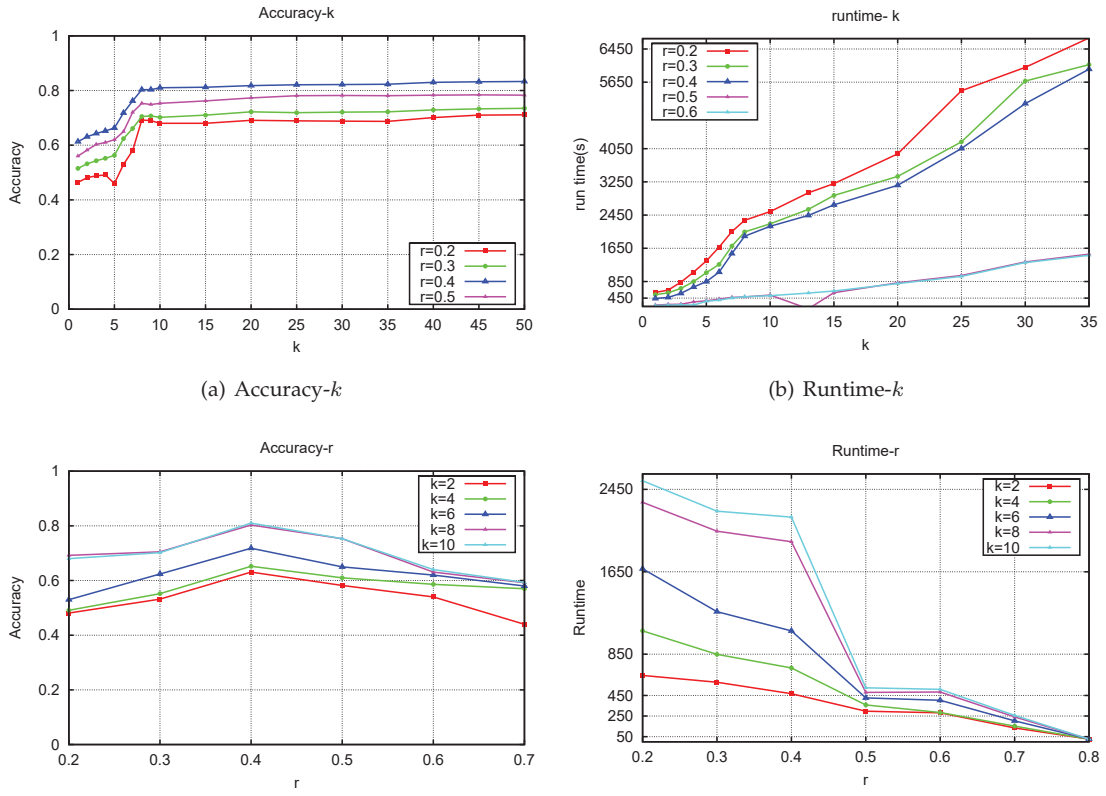



(a) Accuracy-$k$ (b) Runtime-$k$




(c) Accuracy-$r$ (d) Runtime-$r$

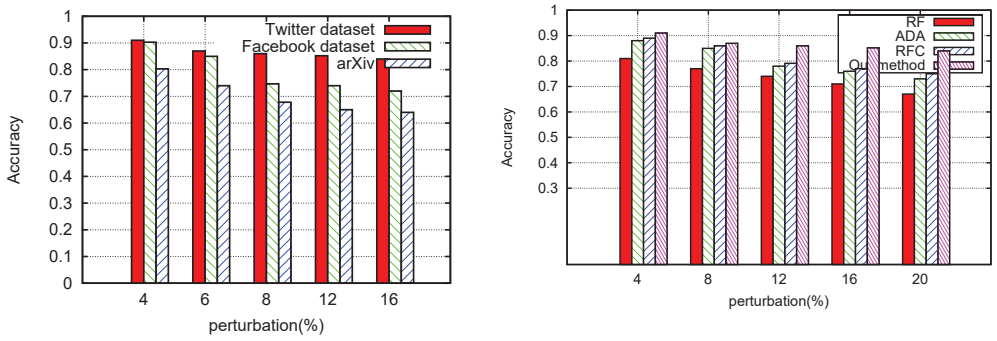
Fig. 4. Experimental results on Facebook dataset with $4\%$ noise



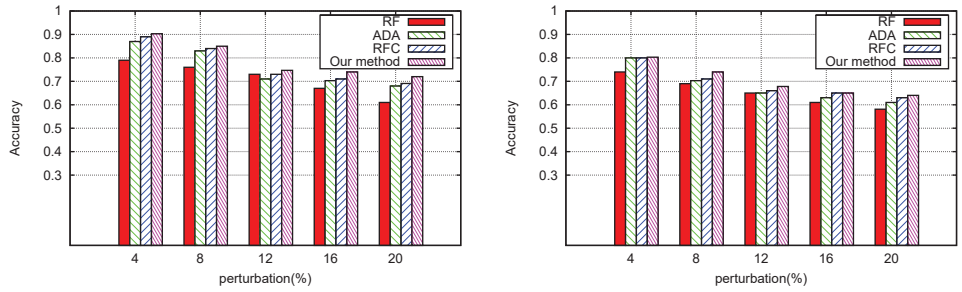

(a) Accuracy-$k$ (b) Runtime-$k$

(c) Accuracy-$r$ (d) Runtime-$r$

Fig. 5. Experimental results on arXiv dataset with $4\%$ noise




(a) Accuracy-perturbation on different datasets (b) Accuracy of different methods on Twitter




(c) Accuracy of different methods on Facebook (d) Accuracy of different methods on arXiv

Fig. 6. De-anonymize results

## REFERENCES


[1] N. B. Ellison. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1): 210–230, 2007.

[2] M. Madejski, M. Johnson, S. M. Bellovin. The failure of online social network privacy settings.*Columbia University Academic Commons, http://hdl.handle.net/10022/AC:P:10666*, 2011.

[3] S. M. Abdulhamid, S. Ahmad, V. O. Waziri. Privacy and National Security Issues in Social Networks: The Challenges. *International Journal of the Computer, the Internet and Management*, 19(3): 14-20, 2014.

[4] T. Song, R. Li, B. Mei, J. Yu, X. Xing, X. Cheng. A Privacy Preserving Communication Protocol for IoT Applications in Smart Homes. *IEEE Internet of Things Journal*, 4(6): 1844-1852, 2017.

[5] X. Zheng, Z. Cai, J. Yu, C. Wang, Y. Li. Follow But No Track: Privacy Preserved Profile Publishing in Cyber-Physical Social Systems. *IEEE Internet of Things Journal*, 4(6): 1868-1878, 2017.

[6] R. Li, T. Song, N. Capurso, J. Yu, J. Couture, X. Cheng. IoT applications on Secure Smart Shopping System. *IEEE Internet of Things Journal*, 4(6): 1945-1954, 2017.

[7] B. Zhou, J. Pei and W. S. Luk. A Brief Survey on Anonymization Techniques for Privacy Preserving Publishing of Social Network Data. In *ACM SIGKDD Explorations Newsletter*, 10(2): 12-22, 2008.

[8] X. Wu, X. Ying, K. Liu. A survey of Privacy-preservation of graphs and social networks. In *Proceedings of Managing and mining graph data*, pp. 421-453, 2010.

[9] M. Hay, G. Miklau, D. Jensen, D. Towsley, P. Weis. Resisting Structural Re-identification in Anonymized Social Networks. In *Proceedings of the VLDB Endowment*, pp. 102-114, 2008.

[10] K. Liu and E. Terzi. Towards Identity Anonymization on Graphs. In *Proceedings of ACM SIGMOD'08*, pp. 1-14, 2008.

[11] C. Dwork. Differential Privacy. In *Proceedings of ICALP*, pp. 1-12, 2006.

[12] N. Li, W. Qardaji, D. Su, Y. Wu and W. Yang. Membership Privacy: A Unifying Framework for Privacy Definitions. In *Proceedings of ACM CSS'13*, pp. 889-900, 2013.

[13] L. Backstrom, C. Dwork, J. Kleinerg. Wherefore Art Thou R3579X? Anonymized Social Networks, Hidden Patterns, and Structural Steganography. In *Proceedings of the 16th International Conference on World Wide Web, ACM*, pp. 181-190, 2007.

[14] A. Narayanan and V. Shmatikov. De-anonymizing Social Networks. In *Proceedings of IEEE Symposiumon Security and Privacy*, pp. 173-187, 2009.

[15] A. Narayanan and V. Shmatikov. Robust De-anonymization of Large Sparse Datasets. In *Proceedings of IEEE Symposium on Security and Privacy*, pp. 111-125, 2008.

[16] G. Wondracek, T. Holz, E. Kirda. A Practical Attack to De-Anonymize Social Network Users. In *Proceedings of IEEE Symposium on Security and Privacy*, pp. 223-238, 2010.

[17] M. Korayem and D. J. Crandall. De-anonymizing users across heterogeneous social computing platforms. In *Proceedings of the 7th international AAAI conference on weblogs and social media*, pp. 1-4, 2013.

[18] M. Srivatsa and M. Hicks. Deanonymizing Mobility Traces: Using Social Networks as a Side-Channel. In *Proceedings of ACM CCS'12*, pp. 628-637, 2012.

[19] S. Ji, W. Li, N. Z. Gong, P. Mittal, and R. Beyah. On your Social Network De-anonymizablity: Quantification and Large Scale Evaluation with Seed Knowledge. In *Proceedings of NDSS*, pp. 1-15, 2015.

[20] S. Nilizadeh, A. Kapadia and Y.-Y Ahn. Community-Enhanced Deanonymization of Online Social Networks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 537-548, 2014.

[21] K. Xing, C. Hu, J. Yu, X. Cheng, F. Zhang. Mutual Privacy Preserving k-Means Clustering in Social Participatory Sensing. *IEEE Transactions on Industrial Informatics*, 13(4): 2066-2076, 2017.

[22] M. Hay, C. Li, G. Miklau. Accurate estimation of the degree distribution of private netowrks. In *Proceedings of ICDM'09*, pp. 169-178, 2009.

[23] M. Hay, V. Rastogi, G. Miklau. Boosting the accuracy of differentially private histograms through consistency, In *Proceedings of the VLDB Endowment*, 3(1-2): 1021-1032, 2010.

[24] E. Zheleva and L. 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, ACM*, pp. 531-540, 2009.

[25] J. Lindamood, R. Heatherly, M. Kantarcioglu, B. Thuraisingham. Inferring private information using social network data. In *Proceedings of the 18th international conference on World Wide Web, ACM*, pp. 1145-1146, 2009.

[26] J. He, W. W. Chu, Z. V. Liu. Inferring privacy information from social networks. In *ISI2006: Intelligence and Security Informatics*, pp. 154-165, 2006.

[27] A. Mislove, B. Viswanath, K. P. Gummadi, P. Druschel. You are who you know: Inferring user profiles in online social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pp. 251-260, 2010.

[28] L. Traud, P. J. Mucha and M.A. Porter. Social structure of facebook networks. *Physical A: Statistical Mechanics and its Applications*, 391(16): 4165-4180, 2012.

[29] A. Chaabane, G. Acs and M. A. Kaafar. You are what you like! Information leakage through users' interests. In *Proceedings of NDSS*, pp. 1-14, 2012.

[30] M. Kosinski, D. Stillwell and T. Graepelb. Private traits and attributes are predictable from digital records of human behavior. In *Proceedings of the National Academy of science of the United States of America*, 110(15): 5802-5805, 2012.

[31] N. Z. Gong and B. Liu. You Are Who You Know and How You Behave: Attribute Inference Attacks via users Social Friends and Behaviors. In *Proceedings of the 25th USENIX Security Symposium*, pp. 979-995, 2016.

[32] S. Ji, W. Li, M. Srivatsa, J. S. He and R. Beyah. Structure based Data Deanonymization of Social Networks and Mobility Traces. In *Proceedings of ISC 2014, Information Security*, pp. 237-254, 2014.

[33] S. Ji, W. Li, M. Srivatsa and R. Beyah. Structural Data De-anonymization: Quantification, Practice, and Implications. In *Proceedings of ACM CCS'14*, pp. 1040-1053, 2014.

[34] J. Qian, X. Y. Li, C. Zhang and L. Chen. De-anonymizing Social Networks and Inferring Private Attributes Using Knowledge Graphs. In *Proceedings of IEEE INFOCOM*, pp. 1-9, 2016.

[35] S. Ji, T. Wang, J. Chen, W. Li. De-SAG: On the De-anonymization of Structure-Attribute Graph Data. *IEEE Transactions on Dependable and Secure Computing*, 99: 1-1, 2017.

[36] M. Naumov, T. Moon. Parallel Spectral Graph Partitioning. In *NVIDIA Technical Report NVR-2016-001*, 2016.

[37] B. Thompson and D. Yao. The union-split algorithm and cluster-based anonymization of social networks. In *Proceedings of ACM Symposium on ICCS'09*, pp. 218-227, 2009.

[38] S. Milgram. The smal world problem. In *Psychology Today*, 1(1): 60-67, May 1967.

[39] Stanford large network dataset collection, https://snap.stanford.edu/data.

[40] arXiv:arXiv bibiography (2016). http://arxiv.org/help/api/index. Accessed 2016.

[41] K. Sharad and G. Danezis, An Automated Social Graph De-anonymization Technique. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, pp. 47-58, 2014.

[42] Jiangtao Ma, Yaqiong Qiao, Guangwu Hu, Yongzhong Huang, Arun Kumar Sangaiah, Chaoqin Zhang. De-anonymizing Soical Networks with Random Forest Classifier. *IEEE Acess*, 6: 10139-10150, 2017.