


RESEARCH

Open Access



Understanding structure-based social network de-anonymization techniques via empirical analysis

Jian Mao^{1*†} , Wenqian Tian^{1,2†}, Jingbo Jiang², Zhaoyuan He², Zhihong Zhou³ and Jianwei Liu¹

Abstract

The rapid development of wellness smart devices and apps, such as Fitbit Coach and FitnessGenes, has triggered a wave of interaction on social networks. People communicate with and follow each other based on their wellness activities. Though such IoT devices and data provide a good motivation, they also expose users to threats due to the privacy leakage of social networks. Anonymization techniques are widely adopted to protect users' privacy during social data publishing and sharing. However, de-anonymization techniques are actively studied to identify weaknesses in current social network data-publishing mechanisms. In this paper, we conduct a comprehensive analysis on the typical structure-based social network de-anonymization algorithms. We aim to understand the de-anonymization approaches and disclose the impacts on their application performance caused by different factors, e.g., topology properties and anonymization methods adopted to sanitize original data. We design the analysis framework and define three experiment environments to evaluate a few factors' impacts on the target algorithms. Based on our analysis architecture, we simulate three typical de-anonymization algorithms and evaluate their performance under different pre-configured environments.

Keywords: Social Network, De-anonymization, Privacy

1 Introduction

Nowadays, social network services have been developed rapidly as a fast-growing business. Social network websites/applications (e.g., Facebook, Youtube, Twitter, Reddit) are getting more and more popular. Users create their personal profiles in a social network platform, sharing their information and interacting with their friends. These activities make social network platforms become huge social data resources, which have great commercial value and significant sociological impacts. Considering the commercial benefit and the social impact of these social network information, the social network service providers may release their social data (consists of users' data) to third parties for academic (e.g., healthcare, social behavior research) or commercial (e.g., market prediction, targeting advertisement) purposes. However, it will

consequently introduce the risk of leaking users' sensitive information (e.g., identity, location, personal interests) [1].

On the other side, the rapid development of wellness smart devices and apps, such as Fitbit Coach and FitnessGenes, has triggered a wave of interaction on social networks. People communicate with and follow each other based on their wellness activities. Though such IoT devices and data provide a good motivation [2], they also expose users to threats due to the privacy leakage of social networks [3].

To protect users' privacy during social data publishing and sharing, the most straightforward solution is to anonymize data by removing users' identities, i.e., *Personally identifiable information*. However, recent research demonstrates that this naïve solution is vulnerable to *auxiliary information-based de-anonymization* [1, 4, 5]. As an improvement, the *edge-editing-based anonymization* scheme is proposed to conceal the social data structure by adding/deleting (Add-Del) and switching edges in the social graph [6]. As a widely adopted approach in relational data anonymization, *k-anonymity* is

*Correspondence: maojian@buaa.edu.cn

†Jian Mao and Wenqian Tian contributed equally to this work.

¹School of Cyber Science and Technology, Beihang University, Xueyuan Road, 100083 Beijing, China

Full list of author information is available at the end of the article

introduced to protect social network data privacy against different attacks. For example, Zhou et al. proposed the k -neighborhood anonymity against neighborhood attack [7]; Liu et al. proposed k -degree anonymity aiming at degree attacks [8]; Zhou et al. [9] aggregated *graph partitioning*, *block alignment*, and *edge copy* techniques and presented k -automorphism to prevent *neighborhood attack*, *degree attack*, *subgraph attack*, and *fingerprint attack* [10, 11]. In addition, anonymization approaches based on aggregation/class/cluster [10, 12, 13], differential privacy mechanisms [14–17], and random walk methods [18] are also proposed to preserve users' private information.

De-anonymization (DA) techniques are actively studied to identify vulnerabilities in current social network data-publishing mechanisms [4]. Typically, these approaches can be classified into two categories, *seed-based de-anonymization* [4, 5, 19–22] as well as *seed-free de-anonymization* [23]. A seed-based de-anonymization approach usually consists of two stages [4]. The first stage is to identify some common (*seed*) users between the anonymized social graph and the auxiliary network graph; the second stage is to conduct de-anonymization propagation iteratively according to social graph structural properties. Nilizadeh et al. proposed community-level-based de-anonymization [19] that can be used to improve other existing seed-based de-anonymization mechanisms [5, 20–23]. Seed-free de-anonymization approaches utilize figure (nodes or edges) properties as the fingerprints and conduct graph matching to de-anonymize the sanitized graph data [23]. Besides, some semantic-based de-anonymization methods are developed to break link privacy [24] or infer private attributes [25]. Ji et al. gave a survey on the graph data anonymization and de-anonymization approaches [26].

There are many factors that influence the implementation performances (e.g., accuracy, scalability) of the existing DA algorithms, for example, the methods (anonymization) sanitizing the original data, parameter configuration, the size of the testing graph, link direction, and density distribution of graph nodes. However, in most existing de-anonymization approaches, they usually aim at one specific anonymization approach or occasionally, do not provide any specification of the methods sanitizing the raw data. Meanwhile, some of them also neglect the evaluation on the de-anonymization accuracy under different parameter configuration.

In this paper, we conduct a comprehensive analysis on the typical structure-based de-anonymization algorithms in social networks to understand the structure-based de-anonymization approaches and disclose the impacts on their application performance caused by

the factors mentioned above. We design the analyzing framework and define three experiment environments to evaluate the factors' impacts on the target algorithms. Based on our analyzing architecture, we simulate three typical de-anonymization algorithms, N-DA scheme [4] proposed by Narayanan et al., Ji-DA scheme [27] proposed by Ji et al., and the structure-attribute graph-based Ji-DeSAG scheme [28], and evaluate their performance under different pre-configured environments.

1.1 Experimental method

Our analyzing architecture includes three key modules, *Anonymization Module*, *De-Anonymization Module*, and *Configuration Module*, as well as two datasets, *original data* and *Anonymized Data*. In the experiments on the first two DA schemes, N-DA and Ji-DA, we use a subset of the *Twitter* social network as our testing dataset, which is one of the most popular social networks. The dataset [29] consists of 90,907 users and 443,399 "follow" relationships. In the third experiment, we use a "Movie" dataset consisting of a star-director-film-writer network [30].

The evaluation results show that for the N-DA scheme, the parameter θ ought to be set as a small value to obtain high accuracy regardless of the anonymization methods and graph data topology. For the Ji-DA scheme, the graph topologies make differences in accuracy, in which depth-spread dataset is weaker to Ji-DA attack. In addition, according to the testing results, the k -degree anonymization method is more vulnerable to this attack. Similarly, the Ji-DeSAG is more efficient to k -degree anonymization method. Moreover, we analyze the three weight parameters and demonstrate that the weight of inheritance similarity is the major impact factor to the accuracy of de-anonymization approaches.

1.2 Contribution

The contributions made by this paper are summarized as follows:

- We make a comprehensive analysis of different dimensions' influences on the accuracy of de-anonymization algorithms.
- We design the analyzing architecture and define three experiment environments. We simulate three typical structure-based de-anonymization algorithms and evaluate their performance under different pre-configured environments.
- Based on our evaluation, we conclude the parameter impacts on the testing approaches and the influences introduced by the topology properties of testing datasets.

1.3 Paper organization

The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 elaborates the background of this paper and gives a brief overview of our work. Section 4 presents the experimental design of the comprehensive analysis on online social network de-anonymization approaches. Section 5 illustrates the evaluation results and corresponding findings. Section 6 concludes the paper.

2 Related work

2.1 Graph anonymization methods

Naïve ID removal is a simple way to anonymized users, and it can keep the best utility of the published data. But it has been proved to be extremely vulnerable to structure-based de-anonymization attacks. Ying et al. [31] developed spectrum preserving randomization methods, Spctr Add/Del and Spctr Switch. The main idea of Spctr Switch (Add/Del) is to randomly switch(add/del) two edges according to the eigenvalues of the graph's adjacency matrix and eigenvalues of the related Laplacian matrix.

Zhou et al. [7] developed an approach to defend the neighborhood attack. Firstly, the neighborhoods of all users are extracted and encoded as minimum depth-first search code, and secondly, group users with similar neighborhoods greedily together and then make any neighborhood in one group $k - 1$ isomorphic neighborhoods in the same group. Liu et al. [32] devised a systematic framework for identity anonymization on graphs. First, a new k -anonymous degree sequence (any degree appears at least k times) is created based on the degree sequence of the original graph. Then, an anonymous graph is constructed based on the k -anonymous degree sequence. Zou et al. [9] proposed a k -automorphic method to protect privacy against all structural attacks conducted before their method and developed a k -match (KM) algorithm to implement the method. A k -automorphic network means that for any vertex of the network, it cannot be distinguished from its $k - 1$ symmetric vertices based on structural information. Similarly, Cheng et al. [33] proposed a k -isomorphic method against structural attacks. They considered both NodeInfo (users identifying information such as names) and LinkInfo (relationships among users). For the k -isomorphism, a graph is divided and anonymized into k subgraphs that are isomorphic.

Hay et al. [10] proposed a partition-level based graph anonymization algorithm which partitioned users and described the graph at the partition level that consisted of supernodes denoting partitions and superedges denoting the density of edges. Thompson et al. [13] presented two new and efficient clustering methods for undirected graphs: bounded t-means clustering and union-split clustering algorithms that divided similar graph nodes into

clusters with a minimum size constraint. Mittal et al. [18] proposed an anonymization based on random walks providing link privacy for perturbing the structure of the social graph in the way that replace the edge (i, j) by another edge (i, u) , where u is the destination of a random walk starting from j .

Sala et al. [14] developed a differentially private graph model called Pygmalion which can preserve as much of the original graph structure as possible, while injecting enough structural noise to guarantee a chosen level of privacy against privacy attacks. Wang et al. [34] developed private dK-graph generation models that enforced rigorous differential privacy in graph generation while preserving utility. Xiao et al. [17] transformed direct edges to connection probabilities via hierarchical random graph (HRG) and inferred the social structure in the sampled HRG model using the Markov chain Monte Carlo (MCMC) method satisfying differential privacy to preserve essential network structural properties.

2.2 Graph de-anonymization attacks

2.2.1 Structural user-based attacks with seed

Backstrom et al. [35] presented the first structural de-anonymization approach in social networks. They used both active and passive attacks in links prediction. As for active attacks, an adversary is able to produce some new "sybil" nodes in the social graph before it is published and to link the "sybil" nodes to those whose privacy him/her wishes to violate. So the adversary makes the sub-graph stand out after original graph being published anonymously. As for passive attacks, colluding adversaries recognize their own sub-graph in anonymized graph which could re-identify users around them. Narayanan et al. [4] proposed a two-stage de-anonymization algorithm of large scale only based on the network topology. The main idea is to re-identify the same node that exists both in the target graph and the auxiliary graph. In the first stage, seed nodes are found in the target graph while using the auxiliary information. In the second stage, seed mappings were used to make large propagation, where the number of commonly mapped nodes was utilized to calculate a similarity score and those who have high scores would be regarded as matches. Then, Narayanan et al. [36] combined de-anonymization with link prediction to de-anonymize the dataset published by Kaggle.com. Simulated annealing-based weighted graph matching is introduced for the seed stage and the propagation stage is promoted into two phases using different threshold to select mappings. In their link prediction, random forests are utilized to make prediction among links which make up for the limitedness of pure de-anonymization. Nilizadeh et al. [19] exploited a divide-and-conquer method to promote de-anonymization algorithms. First, the social structure

is used to make huge networks to several community-level networks and then a two-step graph matching technique is taken into the communities which make the big problem into smaller ones. Nodes mapping is implemented inside communities and then expanded to the whole network.

2.2.2 Structural set-based attacks with seed

Srivatsa et al. [5] developed a de-anonymization attack to mobility traces using social networks as a side channel. Firstly, node betweenness centrality metric is used to find k landmark nodes with the highest scores. Secondly, three methods are presented to the propagation matching, namely, distance vectors, spanning tree matching, and local sub-graph features. Ji et al. [37] discovered a unified similarity-based de-anonymization attack in both social networks and mobility traces. Structural similarity, relative distance similarity, and inheritance similarity are defined and combined to calculate the unified similarity in propagation process.

2.2.3 Structural attacks without seed

Assuming two graphs have the same nodes and start matching the highest degree nodes from the set in two graphs, Ji et al. [38] proposed a seedless cold start optimization-based de-anonymization algorithm. Sharad et al. [39] proposed machine learning-based techniques to de-anonymize nodes in graphs using structural features. They first developed an automated learning model based on neighborhood degree distribution designing a random-forest classifier to predict users' similarity and then presented an end-to-end anonymization attack based on the previous model to re-identify nodes in graphs. Lee et al. [40] proposed a seedless de-anonymization method incorporating multi-hops neighborhood information and

exploiting an improved machine learning technique for matching. Wu et al. [41] provided a systematic study on the effect of overlapping communities on de-anonymization without seed, aiming at minimizing the de-anonymization error.

2.2.4 Structure-attribute-based attacks

Chen et al. [42] utilized user names and the network topology to de-anonymize users. They used user names to reduce the candidates of mapping and then combined the structure information and Levenshtein distance of user names in similarity computing to improve the mapping accuracy. Ji et al. [28] conducted an analysis of attribute-based anonymity on structure-attribute graph (SAG) data and proposed a new de-anonymization framework for SAG data by adding attribute similarity to existent structure-based de-anonymization. Qian et al. [43] utilized knowledge graph to represent arbitrary prior knowledge of attackers and computed the node structural similarity and the attribute similarity to de-anonymization using a knowledge graph. Jiang et al. [44] proposed a de-anonymization scheme based on a structure-attribute framework taking structure characteristics and node properties into consideration, which improved the accuracy of node mapping. Attribute information used in de-anonymization can help obtain better performance. Because it provides more auxiliary information. Besides, users' behavior information are also important in social network privacy inference [45]. In future works, behavior information can be also used as auxiliary information to de-anonymization.

According to the discussion in [26] and our analysis results, we present the comparison of representative structure-based de-anonymization algorithms in Table 1.

Table 1 Comparison of representative de-anonymization methods

DA approach	Vulnerable Anonymization Method			Performance				Category			
	Edge-edit	k-Anony.	U-Split	Para-free	Scalable	Practical	Robust	I	II	III	IV
Backstrom et al. [11]	×	×	×	✓	×	●	×	✓			
N-DA. [4]	✓	●	●	×	✓	✓	✓	✓			
Srivatsa et al.-DV [5]	✓	●	●	✓	●	●	✓		✓		
Nilizadeh et al. [19]	●	●	●	×	●	●	●	✓			
Ji-DA [37]	✓	●	●	×	✓	✓	✓		✓		
Ji-ADA [37]	✓	●	●	×	✓	✓	✓		✓		
Ji-ODA [38]	✓	●	●	×	✓	✓	✓			✓	
Chen et al. [42]	✓	●	●	✓	✓	✓	✓				✓
Ji-DeSAG [28]	✓	●	●	×	✓	✓	✓				✓

As for vulnerability of anonymization methods, ✓ denotes that the attack can succeed under corresponding anonymization methods, × denotes the attack's failure, and ● denotes the attack can conditionally succeed. As for performance, ✓ represents that the attack is capable of corresponding character, × represents the attack's incapability, and ● represents the attack is conditionally capable. Category I: structural user-based attacks with seed. Category II: structural set-based attacks with seed. Category III: structural attacks without seed. Category IV: structure-attribute-based attacks
k-Anony. k -anonymity, *U-Split* union-split, *Para.* parameters

3 Background

In this section, we present the background knowledge related to the de-anonymization analysis. We introduce the general models of social data anonymization and de-anonymization mechanisms and give a brief introduction to several typical structure-based de-anonymization approaches.

3.1 Anonymization models

Users and connections in a social network can be modeled into a graph structure. In this paper, a graph $G = (V, E, W)$ is used to represent a social network, where the node set $V = \{u|u \text{ is a node}\}$ denotes the users, the edge set $E = \{(u, v)|u, v \in V, \text{ and a link exists between } u \text{ and } v\}$ represents the connections among users, and the weight set $W = \{w_{u,v}|u, v \in V, (u, v) \in E, w_{u,v} \in \mathcal{R}\}$ represents the closeness degree of every two connected nodes. If G is an unweighted graph, we just set $w_{u,v} = 1$ for each $(u, v) \in E$.

Before social data are published, they will be sanitized to avoid violation of privacy. The anonymized graph can be modeled as a graph $G^a = (V^a, E^a, W^a)$, where V^a denotes the anonymized users, E^a is the sanitized connections among users, and W^a is the sanitized proximity.

As described above, there are several approaches that an adversary can aggregate auxiliary information about his target. So, we assume that the adversary has collected extra information to de-anonymize the sanitized graph. Similar to anonymized graph, the auxiliary graph denoted as $G^u = (V^u, E^u, W^u)$ (where V^u, E^u, W^u is the users with known identities, known connections, and closeness, respectively) is used to de-anonymize G^a . In order to connect the two graphs, there must be an overlap between G^a and G^u .

3.2 De-anonymization mechanisms

3.2.1 Attack model

Generally, *de-anonymization* is to reidentify an anonymized social graph by using an auxiliary graph, which means to establish a mapping μ between two graphs. We denote the true mapping between two graphs as $\mu_0 : V_{\mu_0}^a \rightarrow V_{\mu_0}^u$. A de-anonymization attack on these two graphs is represented as $\tilde{\mu} : V^a \rightarrow V^u$.

For each $v \in V^a$,

$$\mu(v) = \begin{cases} v', & \text{if } v' = \mu(v) \in V^u; \\ \perp, & \text{if } \mu(v) \notin V^u, \end{cases}$$

where \perp is a *not existing indicator*. After an attack, the outcome mapping is

$$M = \{(v_1, v'_1), (v_2, v'_2), \dots, (v_n, v'_n)\}$$

The mapping is *successful* on $v \in V^a$ when

$$\mu(v) = \begin{cases} \mu_0(v), & \text{if } \mu_0(v) \in V^u; \\ \perp, & \text{if } \mu_0(v) \notin V^u, \end{cases}$$

In this paper, we are trying to evaluate the accuracy of de-anonymization attacks. The accuracy is the fraction of *success* divided by the *size* of anonymized graph.

3.2.2 Structure-based de-anonymization algorithms

Since Backstrom et al. [35] first presented active and passive attacks by creating *sybil* nodes in structural de-anonymization. Here, we introduce the representative algorithms that will be analyzed in the following sections.

Narayanan et al. [4] proposed a *two-stage attack on large-scale propagation* (N-DA). In the first stage, a small number of seeds are found in the target graph instead of many *sybil* nodes. In the second stage, seeds are the key to make large propagation. However, the similarity scores calculated to determine the mappings only consider the information of common nodes so the accuracy is not that high. In addition, the parameter θ (the difference of the max similarity score and the second max similarity score divided by the standard deviation of the mapping set) used in their algorithm was not analyzed in detail, which we prove to have a significant influence on the accuracy of the algorithm.

Ji et al. [37] proposed a *unified similarity-based de-anonymization* (Ji-DA) in both social networks and mobility traces. *Structural similarity*, *relative distance similarity*, and *inheritance similarity* are defined as three similarity metrics to calculate the unified similarity. An adaptive de-anonymization (Ji-ADA) is developed to strengthen the capability of de-anonymization when the overlap between the anonymized graph and the auxiliary graph is very low. The Ji-DA algorithm takes lots of node information, both local graph attributes and whole graph attributes, into consideration. However, the parameters of the method are too many to control without being comprehensively analyzed. They introduce three weight parameters, c_S , c_D , and c_I , to calculate the unified similarity. Besides, the algorithm also includes other parameters, e.g., C as the *similarity loss exponent*, θ as the *de-anonymization threshold*, and ϵ as the *mapping control factor*. Coincidentally, all these parameters influence the accuracy of the method respectively.

Ji-DeSAG [28] is a de-anonymization algorithm combining graph structure and attribute information of users in the social network based on the structure-attribute graph (SAG) model. In the SAG model, the attributes are represented by nodes and links between attribute nodes and user nodes represent the belonging of attributes to corresponding users. It combines user-based structural de-anonymization and set-based structural de-anonymization techniques. Due to the dependency to the structure-based DA approach, its accuracy is influenced by the factors related to structure-based de-anonymization, e.g., similarity rate S_a and weighting parameter c .

In the next section, we present our design to evaluate the de-anonymization capabilities of those algorithms corresponding to the selected influential factors.

4 Design and implementation

As we analyzed previously, there are two critical factors that influence the efficiency of de-anonymization algorithms, anonymization method and parameter configuration. In this section, we comprehensively analyze the efficiency of the typical de-anonymization algorithms with respect to these two influential factors, namely, different kinds of anonymization methods and different preferences of significant parameters. We present the overall architecture in Fig. 1. As shown in Fig. 1, our analyzing architecture includes three key modules, *anonymization module*, *de-anonymization module*, and *configuration module*, as well as two datasets, *original data* and *anonymized data*. We illustrate the experiment design details of these components in the following subsections, respectively.

Our evaluation scheme targets graph data and algorithms. As the seed production stage is not our priority, we mainly focus on the propagation stage. In this paper, seed production can be implemented in the same way as existing solutions [4, 5, 35, 36]. So, in the following experiments, we assume we have selected k seed mappings, denoted by $M_s = \left\{ (s_1, s'_1), (s_2, s'_2), \dots, (s_k, s'_k) \right\}$, where $s_i \in V^a, s'_i \in V^u$, and $s'_i = \mu(s_i)$.

4.1 Dataset preparation

As the first two DA algorithms (N-DA, Ji-DA) only use network structure information and the third DA algorithm (Ji-DeSAG) use both structure and attribute information, we use two different datasets (shown in Table 2) in our evaluation. In the experiments on the first two DA schemes, N-DA and Ji-DA, we use a subset of the Twitter social network as our evaluation input, which is one of the most popular social networks nowadays. The dataset [29] consists of 90,907 users and 443,399 “follow” relationships. And in the third experiment, we use the

Table 2 Graph properties of Twitter dataset

Network	Nodes	Edges	Max.Deg	Av.Deg
Twitter	90907	443399	230	9.755
Movie	12285	61962	1501	11.125

Max.Deg represents the maximum node’s degree, *Av.Deg* represents the average node’s degree

“Movie” dataset consists of a star-director-film-writer network [30]. It consists of 12285 nodes and 61962 edges shown in Table 2 that contains both network structure and attribute information, which are necessary for our analysis.

As the *Twitter* dataset is a huge directed graph, before testing, we first divide the Twitter-network into several sub-graphs to achieve better evaluation results and turn them into both undirected or directed sub-graphs according to our experiments. We use a *center-spread* method to obtain several smaller subsets for experiments. And we process the raw/original data in different ways with regard to directed and undirected graphs. The dataset splitting method is listed as follows. We divide the Twitter-network into several parts to achieve better evaluation results.

- Step 1: When undirected subsets are required, we transfer the original directed network into an undirected graph by using the approach mentioned in the work [46], keeping the edge that only exists bilaterally. If directed subsets are required, we directly go to the next step.
- Step 2: Select m max-degree nodes $\{v_1, v_2, \dots, v_m\}$ in the original graph G (Twitter dataset) and put them in the top-degree set denoted as $Tset$.
- Step 3: Let each $v_i \in Tset$ be the center. Select the neighbors (both in_edge neighbor and out_edge neighbor for the directed graph) of v_i and add them into $Tset$.
- Step 4: Repeat the Step 3 for n times and obtain a subset $graph_m - n$ for an undirected graph and $Digraph_m - n$ for a directed graph.

Table 3 presents the subsets obtained by following Step 1–Step 4.

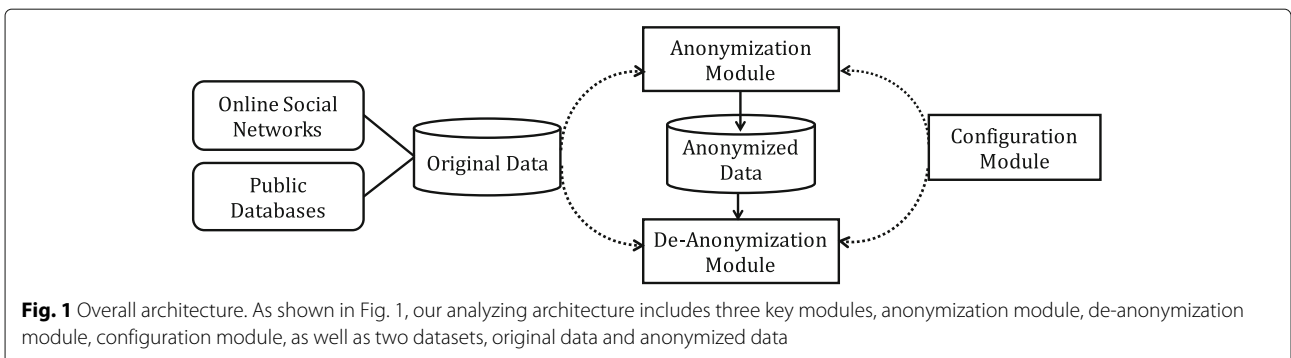


Fig. 1 Overall architecture. As shown in Fig. 1, our analyzing architecture includes three key modules, anonymization module, de-anonymization module, configuration module, as well as two datasets, original data and anonymized data

Table 3 Sub-figures deduced from Twitter dataset

Raw network	Nodes	Edges	Max.Deg	Av.Deg
<i>graph_1</i> – 3	416	785	96	3.774
<i>graph_2</i> – 2	601	1029	96	3.424
<i>Digraph_1</i> – 3	1463	3639	230	4.975
<i>Digraph_2</i> – 2	767	3032	230	7.906

These subsets listed in Table 3 are used as auxiliary graphs to re-identify anonymized graphs. To analyze the structure-based de-anonymization mechanisms thoroughly, we introduce several anonymized methods that we use to prepare sanitized social datasets in our experiments.

4.2 Selected anonymization algorithms

4.2.1 Naïve add/del edges method

For naïve *edge-edit* anonymized method, we choose add/del edges method [31] as one of our sanitized approaches, which protect node and link privacy of graph data by adding or deleting edges randomly through the whole graph. We use this method to anonymize all the datasets with different sizes. When using it, we set the fraction of edges that we want to edit. For instance, if we set the edition fraction as 0.1, actually, the overlap of edges we get will be lower than 0.9. After a subset *graph_m – n* (*Digraph_m – n*) being anonymized by add/del edges method, it is denoted as *graph_m – n_{add} – del* (*Digraph_m – n_{add} – del*). The processed subset samples are listed in Table 4.

4.2.2 *k*-degree anonymization method

As we discussed in Section 3, *k*-anonymity-based solutions are also a typical choice for preserving social data privacy. In this paper, we select a representative variant of the *k*-anonymity based method, that is *k*-degree anonymization [32], as a candidate algorithm for social data sanitization/pre-processing. It works as that for every node there exist at least $k - 1$ nodes with same degree in the graph. In the algorithm, we set the different *k* to get different anonymized graphs with different overlaps. We denote a subset *graph_m – n* (*Digraph_m – n*) being anonymized by *k*-degree anonymization method as *graph_m – n_{kda}* (*Digraph_m – n_{kda}*) in this paper. The preprocessed subset samples are listed in Table 5.

Table 4 Subset samples preprocessed by naïve add/del edges method

Anonymized network	Nodes	Edges	Edge overlap
<i>graph_1</i> – 2 _{add} – del	196	431	0.82
<i>Digraph_1</i> – 2 _{add} – del	379	2384	0.61

Table 5 Subset samples preprocessed by the *k*-degree anonymization method

Anonymized network	Nodes	Edges	Edge overlap
<i>graph_1</i> – 2 _{kda}	196	458	0.82
<i>Digraph_1</i> – 2 _{kda}	379	2548	0.61

4.2.3 Union-split method

The cluster based methods [10, 47, 48] are similar to the *k*-anonymity-based methods. The aim is to make nodes in a cluster indistinguishable on structure. There are several approaches to implement it, such as *t*-means [47] and union-split [47]. In this paper, we use union-split method to anonymize graphs, denoted as *graph_m – n_{union}* (*Digraph_m – n_{union}*). Samples are listed in Table 6.

We use these three anonymization approaches in our evaluation. The target de-anonymization algorithms are described in the following subsections.

4.3 Target de-anonymization algorithms

As we discussed in Section 2, there are four types of de-anonymization methods. Because the influential factors on performance of the de-anonymization method without seed are similar to those with seed, in this paper, we focus on three types of seed-based DA algorithms (structural user-based attacks with seed, structural set-based attacks with seed, and structure-attribute-based attacks). We select one representative algorithm in each type to analyze comprehensively and test their de-anonymizability. In this paper, we mainly focus on the propagation step, so in all the algorithms we test, we will take pre-selected seed mappings M_s as input.

4.3.1 Narayanan et al. de-anonymization (N-DA)

N-DA [4] is a classic de-anonymization algorithm and also a milestone in the field of de-anonymization researches. It is efficient to the naïve anonymization methods and also is the basis of many other follow-up approaches. As a result, it is important to analyze this algorithm for better understanding of structural attacks.

The N-DA algorithm [4] takes two directed graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ and seed mappings M_s as inputs. It outputs a mapping μ . In the propagation stage, for each iteration, it picks an unmapped node $v \in V_1$ and calculates a *score* for each $(v, v'), v' \in V_2$. The mappings between (v, v') with a score over a threshold will remain. When switching the two input graphs, if v' maps back to v , then the mapping between v and v' will be

Table 6 Subset samples preprocessed by union-split method

Anonymized network	Nodes	Edges	Edge overlap
<i>graph_1</i> – 2 _{union}	196	407	0.82
<i>Digraph_1</i> – 2 _{union}	379	2230	0.61

added to the output mapping list. The propagation dose not converge until no more mappings can be added to the final list. The *score* above equals to the number of common nodes of v and v' that have been mapped. In this algorithm, *eccentricity* in [4] equals the difference of the maximum similarity score and the second maximum similarity score divided by the standard deviation of the mapping set. If the *eccentricity* of the match scores is bigger than the threshold θ , the mapping (v, v') with the maximum score can be added to the final mapping list. θ is an important parameter that influences the output accuracy greatly. In this paper, we will analyze the parameter in different angles.

4.3.2 Ji et al. de-anonymization (Ji-DA)

Ji-DA [37] is a most recent de-anonymization approach, which is built upon the strength of several previous work and aggregates a large amount of graph topology information. The evaluation can help to understand how the graph topology contributes to de-anonymization approaches.

Ji-DA algorithm takes two undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ and seed mappings M_s as input. The output is the mapping between these two graphs. For each iteration, it starts from the neighbors of the already mapped nodes M and calculates a *unified similarity score* $s(v, v')$ between every pair of $\{v | v \in V_1 \& v \notin M\}$ and $\{v' | v' \in V_2 \& v' \notin M\}$ to construct a weighted bipartite graph B based on $s(v, v')$. It uses the Hungarian algorithm to obtain a *maximum weighted bipartite matching* M' of B . A *threshold* and a *TOP-K* strategy are used to remove some improper mappings. Finally, the remained mappings are added into the mapping M . The whole algorithm contains many parameters. In our experiment, we especially pay attention to the impacts on the similarity score s caused by three parameters c_S , c_D , and c_I , where c_S , c_D , and c_I represent the weights of structural similarity, relative distance similarity, and inheritance similarity, correspondingly.

4.3.3 Ji et al. structure-attribute-based de-anonymization (Ji-DeSAG)

Ji-DeSAG [28] is based on both user-based structural de-anonymization and set-based structural de-anonymization. In the SAG model, the attributes are represented by nodes and links between attribute nodes and user nodes represent the belonging of attributes to

Table 7 Representative algorithms and parameters

Algorithm	Network topology	Parameter
N-DA	Directed graph	θ
Ji-DA	Undirected graph	$c_S, c_D, c_I, C, \theta, \epsilon$ etc.
Ji-DeSAG	(Un)directed graph	c

Table 8 Sample subsets selected according to depth-spread

Raw Network	Nodes	Edges	Av.Deg
<i>graph</i> ₁ – 2	196	431	4.398
<i>graph</i> ₁ – 3	416	785	3.774
<i>graph</i> ₁ – 4	1062	1680	3.164
<i>graph</i> ₁ – 5	3599	5905	3.281
<i>Digraph</i> ₁ – 2	379	1739	9.177
<i>Digraph</i> ₁ – 3	1463	3639	4.975
<i>Digraph</i> ₁ – 4	7036	15041	4.275

corresponding users. The Ji-DeSAG algorithm is based on the previous structure-based DA algorithms but the attribute similarity is added to the similarity score computed between nodes. Both user-based structural DA and set-based structural DA can be extended to the Ji-DeSAG algorithms. In this paper, we take the user-based structural DA promotion as an example to analyze. It takes two directed graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ and seed mappings M_s as input. It outputs a mapping μ . In the propagation stage, for each iteration, it calculates a similarity score S for each unmapped (v, v') , $v \in V_1$ and $v' \in V_2$. S is determined by the attribute similarity S_a and the structure similarity S_s . S_a is equal to one minus the attribute difference between two nodes divided by the max attribute difference, and S_s is the same as the previous DA algorithm. The similarity rate S is calculated as $S = c * S_s + (1 - c) * S_a$, where c is a weighing parameter that balances the weight between attribute similarity and structure similarity.

4.4 Critical factors and experiment design

In this subsection, we choose the methods in [4, 28, 37] as our target algorithms for analysis. In these selected approaches [4, 28, 37] and other recent de-anonymization attacks, the number of seeds and noise proportion are two general pre-conditions for researchers evaluating the efficiency of their approaches. In this paper, besides these

Table 9 Sample subsets selected according to width-spread

Raw network	Nodes	Edges	Av.Deg
<i>graph</i> ₁ – 2	196	431	4.398
<i>graph</i> ₂ – 2	601	1029	3.424
<i>graph</i> ₃ – 2	753	1204	3.293
<i>graph</i> ₄ – 2	981	1578	3.217
<i>Digraph</i> ₁ – 2	379	1739	9.177
<i>Digraph</i> ₂ – 2	767	3032	7.906
<i>Digraph</i> ₃ – 2	1096	7702	14.055

Table 10 Datasets used for N-DA

Original graph	Anonymized graph	Edge overlap
Digraph ₁ – 2	Digraph ₁ – 2 _{add – del}	61%
	Digraph ₁ – 2 _{kda}	
	Digraph ₁ – 2 _{union}	
Digraph ₁ – 3	Digraph ₁ – 3 _{add – del}	58%
	Digraph ₁ – 3 _{kda}	
	Digraph ₁ – 3 _{union}	
Digraph ₂ – 2	Digraph ₂ – 2 _{add – del}	63%
	Digraph ₂ – 2 _{kda}	
	Digraph ₂ – 2 _{union}	

two factors, we analyze the DA algorithms in the following aspects.

4.4.1 Algorithm parameter configuration

Most algorithms have one or more parameters, and these parameters often have great effects on the accuracy of de-anonymization. We will analyze several key preferences in [4], [37], and [28]. For the N-DA scheme, we analyze the influence of accuracy with regard to the eccentricity θ in different angles. We choose some directed subsets we described above and use different anonymization methods. In the Ji-DA scheme, we analyze the effect of accuracy with respect to three weighing factors c_S , c_D , and c_I to observe the connections among them. For the Ji-DeSAG scheme, we analyze the influence caused by the parameter c . The selected critical parameters are shown in Table 7.

4.4.2 Topology properties of the social data

When it comes to graph de-anonymization, the graph structure is bound to influence the accuracy. As we described in the previous part, we have created many subsets of graph data by using a *center-spread* method. Actually, there will be two ways of spreading. One is *depth-spread* and the other is *width-spread*. So, we will obtain two types of subsets. In this paper, we use both types to analyze the de-anonymization methods.

Depth-spread In our experiments, we fix the center nodes and expand to their neighbors in deep-level. The selected subsets are listed in Table 8.

Width-spread In this paper, we fix the hops of neighbors and choose different number of nodes as center nodes in width-level. The subsets are shown in Table 9, and the corresponding evaluation results are illustrated in Section 5.

4.4.3 Performance metrics

There are several metrics to evaluate the degree of the re-identification. *Accuracy* is the successful rate of final matches, which equals to the number of successful matches divided by the number of mutually existing nodes in targeting graph. Another metrics is *recall rate*, which is the proportion of correct matches divided by the number of nodes existing in both graphs. The error rate and precision are also used to quantify the de-anonymization results. In addition, *receiver operating characteristic (ROC) curve* is selected to measure the

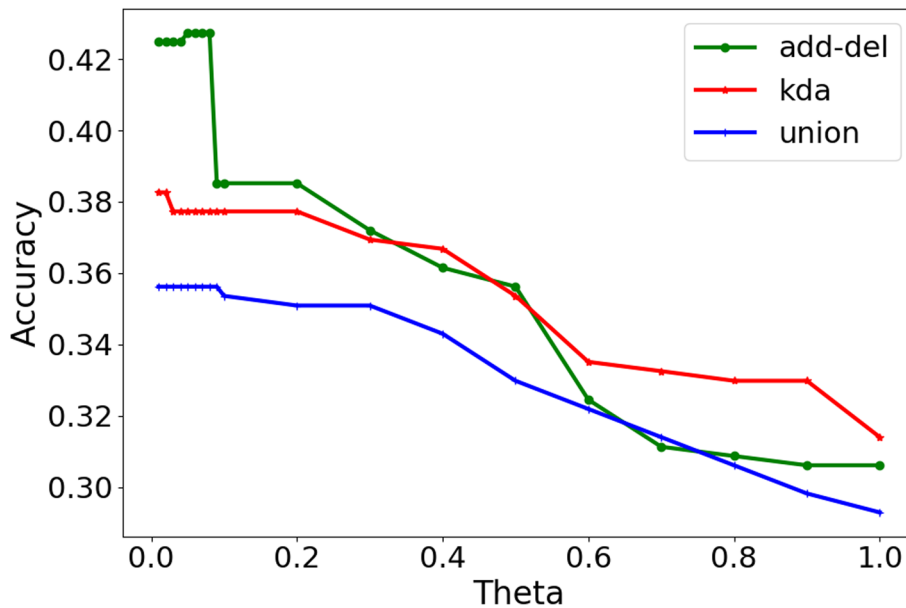


Fig. 2 θ 's accuracy impact in Digraph₁–2. The results of θ 's accuracy impact with respect to three anonymization methods in different topologies in Digraph₁–2

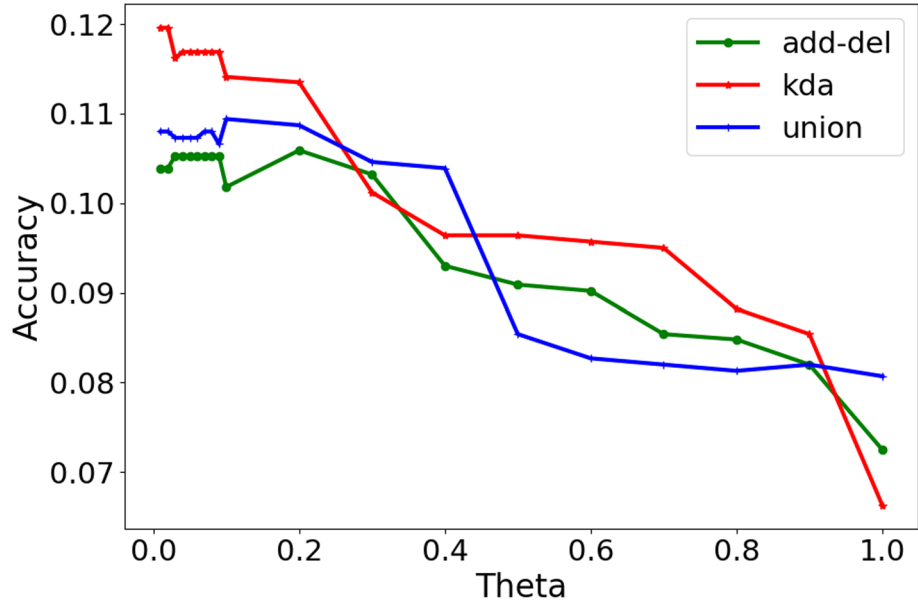


Fig. 3 θ 's accuracy impact in Digraph_1-3. The results of θ 's accuracy impact with respect to three anonymization methods in different topologies in Digraph_1-3

identification performance considering the true positive and false positive simultaneously. In most existing de-anonymization researches, accuracy is usually the first consideration. Accordingly, in this paper, we use *accuracy* as the metrics to evaluate the de-anonymization algorithm performance.

5 Evaluation

In this section, we evaluate different de-anonymization approaches described in Section 4 via three different anonymization methods with several important angles we mentioned above. In the following part, we will illustrate the evaluation on each de-anonymization algorithm

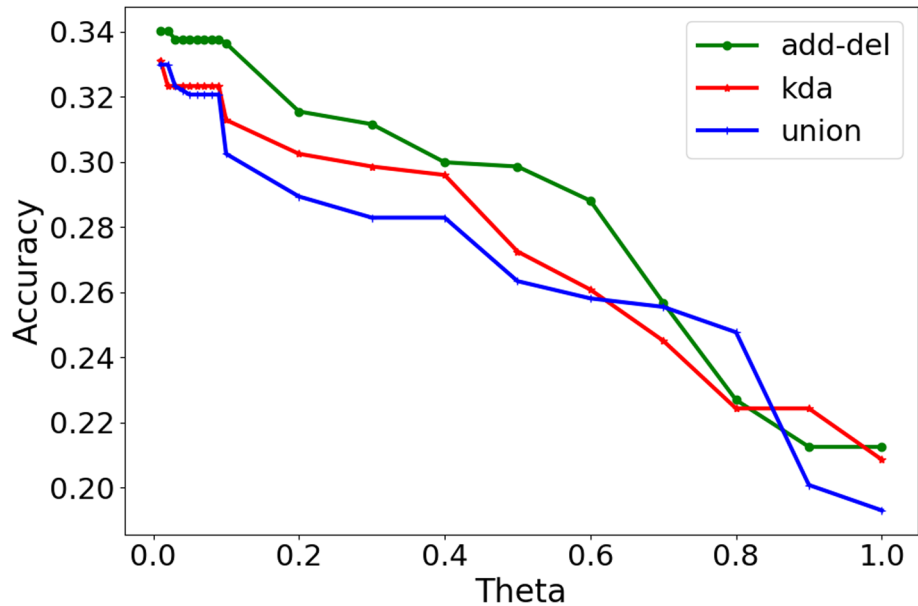


Fig. 4 θ 's accuracy impact in Digraph_2-2. The results of θ 's accuracy impact with respect to three anonymization methods in different topologies in Digraph_2-2

Table 11 Datasets used for the first experiment of Ji-DA

Original graph	Anonymized graph	Edge overlap
graph ₁ – 2	graph ₁ – 2 _{add-del}	82%
	graph ₁ – 2 _{kda}	
	graph ₁ – 2 _{union}	
graph ₁ – 3	graph ₁ – 3 _{add-del}	82%
	graph ₁ – 3 _{kda}	
	graph ₁ – 3 _{union}	
graph ₁ – 4	graph ₁ – 4 _{add-del}	82%
	graph ₁ – 4 _{kda}	
	graph ₁ – 4 _{union}	
graph ₂ – 2	graph ₂ – 2 _{add-del}	82%
	graph ₂ – 2 _{kda}	
	graph ₂ – 2 _{union}	
graph ₄ – 2	graph ₄ – 2 _{add-del}	82%
	graph ₄ – 2 _{kda}	
	graph ₄ – 2 _{union}	

respectively. All the datasets of the following experiments are based on the subsets we configured in Section 4.

Our simulation is conducted on a computer with an Intel Core 3.2 GHz processor and 4 GB RAM.

5.1 Experiment analysis of N-DA

In the experiment on N-DA approach, we evaluate the influence of θ on the accuracy of N-DA algorithm under different anonymization methods and graph topologies. As the seed identification stage is not our primary

purpose, we directly set 50 top-degree nodes as seeds. To analyze the effects of different anonymization methods regarding to one subset, we set the edge overlap between the anonymized graph and the auxiliary graph to be same. For simplicity, the node overlaps in our experiment are set as 1, i.e., the edge overlap between *Digraph₁ – 2* and *Digraph₁ – 2_{add-del}*, *Digraph₁ – 2_{kda}*, *Digraph₁ – 2_{union}* is the same. The datasets we used are listed in Table 10.

The results of θ 's accuracy impact with respect to three anonymization methods in different topologies are shown in Figs. 2, 3, and 4, respectively.

Observation. The experiment results demonstrate that regardless of the dataset topology or anonymization algorithms, the accuracy goes down with the parameter θ getting higher. So, a lower value of the parameter θ will contribute to a higher re-identification accuracy. And as the depth-spread dataset gets much lower accuracy than the width-spread dataset, depth-spread graphs seem to be more vulnerable to the N-DA algorithm.

5.2 Experiment analysis of Ji-DA

In this subsection, we evaluate the influence of three key parameters, different anonymization methods, and graph topologies on accuracy of Ji-DA.

We set 30 top-degree nodes as seeds and evaluate the influence of different anonymization methods and graph topologies. In this experiment, the parameters of the Ji-DA algorithm are set as follows: $C = 0.9$, $c_S = 0.2$, $c_D = 0.6$, $c_I = 0.2$, $\theta = 0.9$, $\delta = 1$, and $\epsilon = 0.5$. The selected

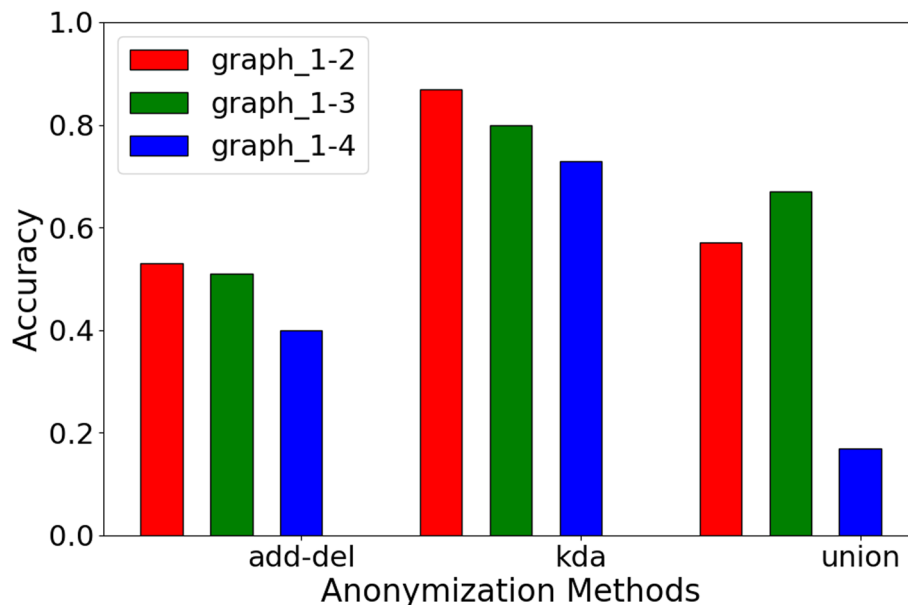


Fig. 5 Effect of the anonymization method on accuracy for depth-spread subsets. The effect of the anonymization method on accuracy for depth-spread subsets

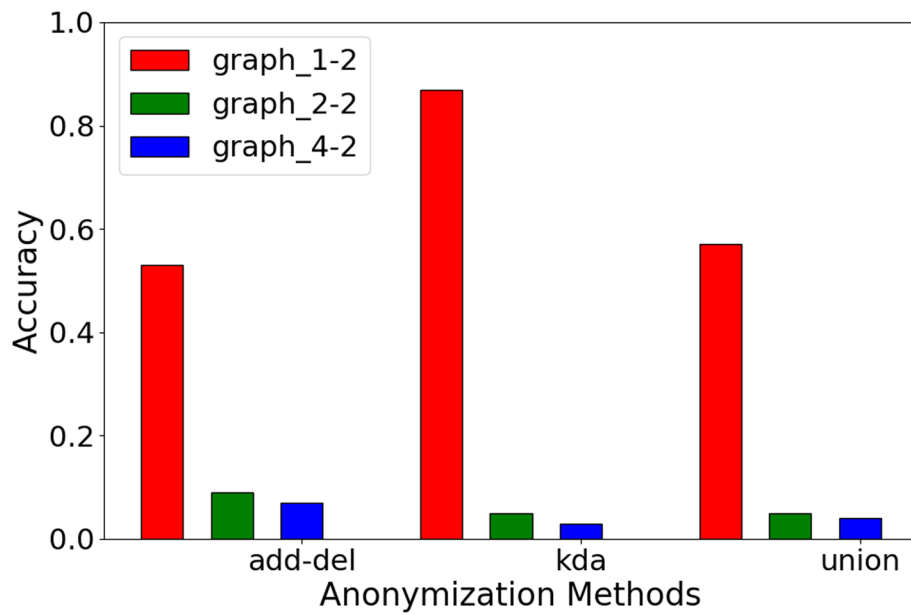


Fig. 6 Effect of the anonymization method on accuracy for width-spread subsets. The effect of the anonymization method on accuracy for width-spread subsets

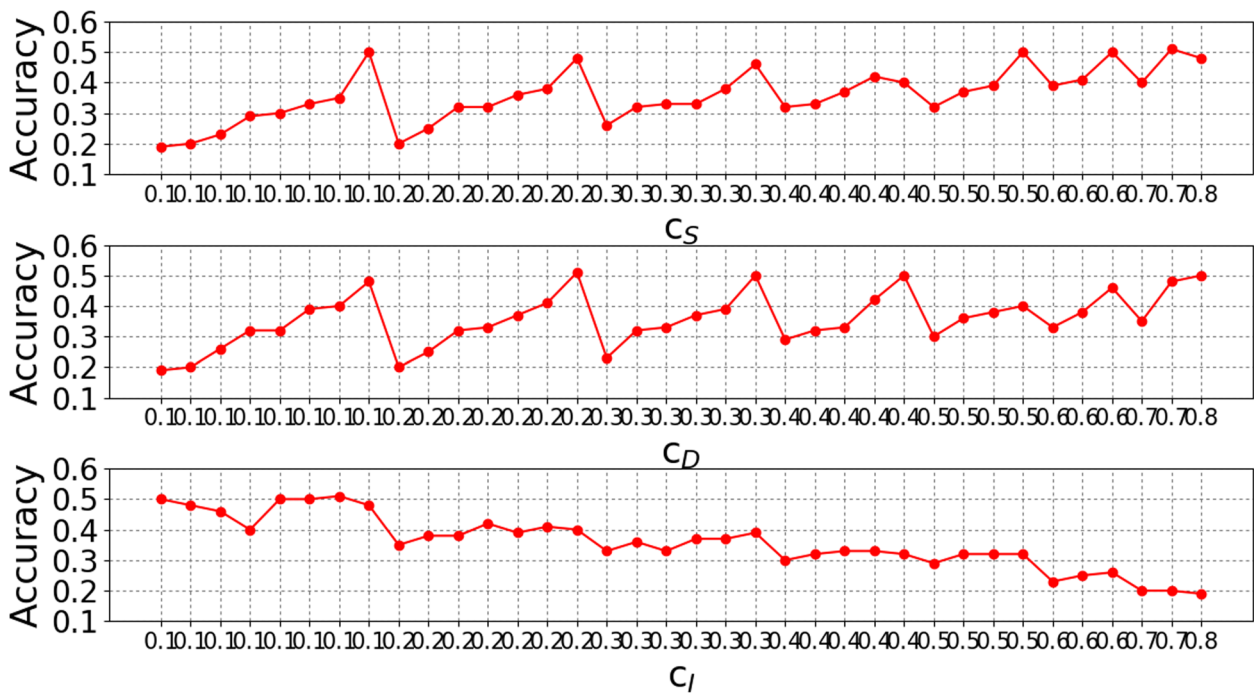
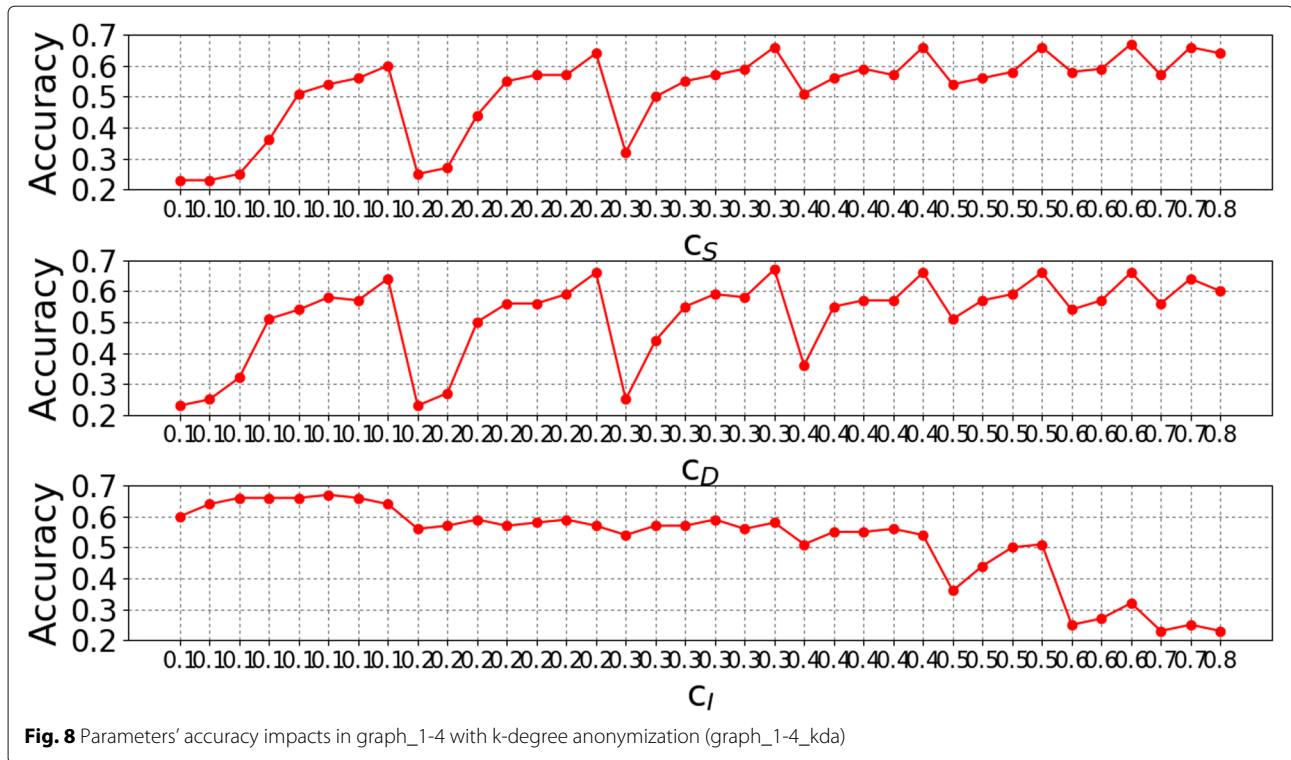


Fig. 7 Parameters' accuracy impacts in graph_1-4 with add/del anonymization (graph_1-4_add-del)



datasets are listed in Table 11. The accuracy impact results of different anonymization and graph topologies are shown in Figs. 5 and 6.

We choose the subset *graph_1 – 4* with *graph_1 – 4_add – del* and *graph_1 – 4_kda* (in which the edge overlap is 82%) as anonymized graphs to analyze the three weighing parameters c_S , c_D , and c_I . To set different preferences, we set the full permutation of three parameters that each value of the parameter varies from the interval [0.1, 0.8] whose increasing step is 0.1. Besides, the sum of three parameters c_S , c_D , and c_I is 1. For example, $c_S = 0.1$, $c_D = 0.1$, $c_I = 0.8$; $c_S = 0.2$, $c_D = 0.5$, $c_I = 0.3$. There are 36 groups of parameter settings of these three parameters. The other parameters are configured as: $C = 0.9$, $\theta = 0.9$, $\delta = 1$, and $\epsilon = 0.5$.

The results of three key parameters regarding to subset *graph_1 – 4* with *graph_1 – 4_add – del* anonymization method are shown in Fig. 7. The results of three key parameters regarding to subset *graph_1 – 4* with *graph_1 – 4_kda* anonymization method are shown in Fig. 8. As we do not optimize the other parameter such as θ and the overlap between two graphs is small, the accuracy may be a little bit low. Nevertheless, our purpose here is not to maximize the accuracy but to analyze the influence tendency of the three weighing parameters.

Observation. In the first experiment (anonymization method and subset topology), the accuracy of depth-spread graphs is much higher than that of the width-spread graphs. We check the two groups of datasets,

finding the edges of depth-spread graphs are more than the edges of width-spread graphs when their nodes are the same, showed in Table 12. And if the value of $\frac{Edges}{Nodes}$ is bigger, the accuracy is higher. We think the Ji-DA is more suitable to attack the networks that have much more edges than nodes. In addition, based on the testing results, the k-degree anonymization method is more vulnerable to this attack.

In the second experiment (three weighing parameters), we find that the accuracy goes down when c_I gets bigger, while c_S and c_D seem to present period tendency. In [37], c_S represents the *structural similarity* of a node, which considers the node's global information and c_D represents the *relative distance similarity* of a node, which also considers the nodes's global information partly. So, the two parameters have some common meanings. However, the c_I represent the *inheritance similarity* of a node, which considers the node's nearby neighborhoods that

Table 12 Nodes and edges of depth and width datasets

Original graph	Nodes	Edges	$\frac{Edges}{Nodes}$
<i>graph_1 – 2</i>	196	431	2.199
<i>graph_1 – 3</i>	416	785	1.887
<i>graph_1 – 4</i>	1062	1680	1.582
<i>graph_1 – 2</i>	196	431	2.199
<i>graph_2 – 2</i>	1466	2278	1.554
<i>graph_4 – 2</i>	1844	2827	1.533

Table 13 Datasets used for the third experiment of Ji-DeSAG

Original graph	Anonymized graph	Edge overlap
Movie	<i>Movie_add – del</i> <i>Movie_kda</i>	48%

have mapped so it is different from the previous two parameters. Accordingly, we consider that the parameter c_I may influence the re-identification accuracy of this algorithm greatly. A small value of c_I contributes to a high accuracy.

5.3 Experiment analysis of Ji-DeSAG

In this subsection, we evaluate the influence of critical parameter c on the accuracy of Ji-DeSAG with respect to different anonymization methods (naïve add/del edges anonymization and k -degree anonymization). We use “Movie” dataset to evaluate the performance, and the configuration is shown in Table 13. As we mentioned in the previous section that the Ji-DeSAG algorithm is based on the structure-based DA; here, we evaluate it based on the structural user-based DA. We improve N-DA by adding nodes’ attributes similarity when the similarity score is computed. We set 50 top-degree nodes as seeds and $\theta = 0.0000001$. We set different c to test the accuracy. The experiment result is shown in Fig. 9.

Observation. We can see from the figure that when c is smaller than 0.1, the accuracy goes up with c getting bigger. But when c is bigger than 0.1, the accuracy goes down slowly with c getting stable and then getting bigger. So, presence of the attribute similarity S_a contributes to the performance of the DA algorithm and the weight of S_a cannot be a big value. And the accuracy of groups using k -degree anonymization method is higher than groups using the naïve add/del-edge method, which means Ji-DeSAG is more efficient to the k -degree anonymization method.

5.4 Results and discussion

According to the evaluation results, we come up to the following conclusions.

For the N-DA scheme, the parameter θ ought to be set much lower to obtain high accuracy regardless of the anonymization methods and the graph data topology. For Ji-DA, the results show that the graph topology makes a difference in accuracy, where the depth-spread dataset will be more vulnerable to the attack. In addition, based on the testing results, the k -degree anonymization method is more vulnerable to this attack. Moreover, we analyze the three weighing parameters, which is shown that the weight of inheritance similarity c_I is the major factor influencing the de-anonymization accuracy. As the accuracy goes down with c_I increasing while the other parameters seem to have a period trend along with c_I . For Ji-DaSAG, the attribute similarity S_a contributes to the performance of the DA algorithm and the weight of S_a cannot be a big value. And the Ji-DeSAG is more efficient to the k -degree anonymization method.

6 Conclusion

In this paper, we conduct a comprehensive analysis on the typical structure-based social network de-anonymization algorithms to achieve a deep understanding on the de-anonymization approaches and disclose the impacts on their application performance caused by the factors mentioned above. We design the analyzing framework and define three experiment environments to evaluate the factors’ impacts on the target algorithms. Based on our framework, we simulate three typical de-anonymization algorithms and evaluate their performance under different pre-configured environments.

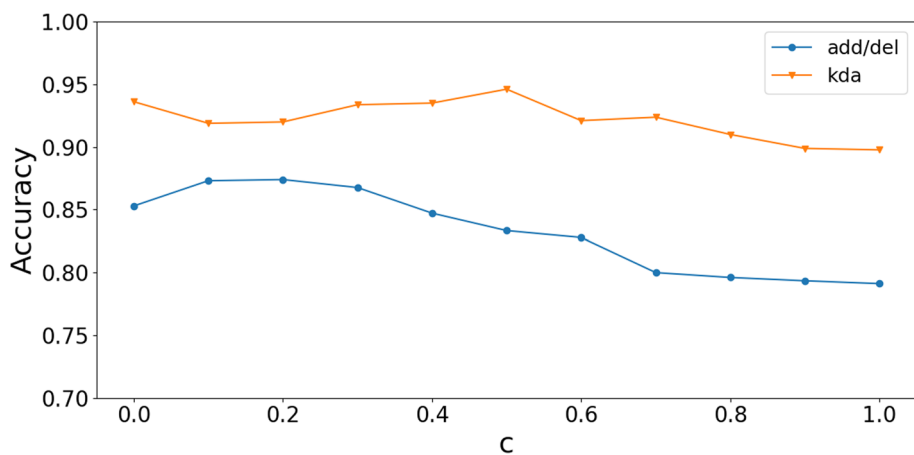


Fig. 9 Effect of weighing parameters c on accuracy of Ji-DeSAG with different anonymization methods. In Figure 9, when c is smaller than 0.1, the accuracy goes up with c getting bigger. But when c is bigger than 0.1, the accuracy goes down slowly with c getting stable and then getting bigger

Abbreviations

ADA: Adaptive de-anonymization; Add-del: Adding/deleting; DA: De-anonymization; IoT: Internet of Things; Ji-DA: Ji et al. de-anonymization; Ji-DeSAG: Ji et al. structure-attribute-based de-anonymization; N-DA: Narayanan et al. de-anonymization; ROC: Receiver operating characteristic; SAG: Structure-attribute graph

Acknowledgements

Not applicable.

Funding

This work was supported in part by the National Key R&D Program of China (No. 2017YFB0802400), the National Natural Science Foundation of China (No. 61402029, No. 61379002, No. U11733115), and the Funding Project of Shanghai Key Laboratory of Integrated Administration Technologies for Information Security (No. AGK201708).

Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Authors' contributions

The authors have contributed jointly to the manuscript. All authors have read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

¹School of Cyber Science and Technology, Beihang University, Xueyuan Road, 100083 Beijing, China. ²School of Electronic and Information Engineering, Beihang University, Xueyuan Road, 100083 Beijing, China. ³Laboratory of Integrate Administration Technologies for Information Security, Shanghai Jiao Tong University, Shanghai, China.

Received: 13 September 2018 Accepted: 7 November 2018

Published online: 05 December 2018

References

1. A. Narayanan, V. Shmatikov, in *Security and Privacy, 2008. SP 2008. IEEE Symposium On*. Robust de-anonymization of large sparse datasets (IEEE, New York, 2008), pp. 111–125
2. X. Zheng, Z. Cai, Y. Li, Data linkage in smart IoT systems: a consideration from privacy perspective. *IEEE Commun. Mag.* **56**(9), 55–61 (2018)
3. Fitness app Strava exposes the location of military bases. <https://techcrunch.com/2018/01/28/strava-exposes-military-bases/>. Accessed 26 July 2018
4. A. Narayanan, V. Shmatikov, in *Security and Privacy, 2009 30th IEEE Symposium On*. De-anonymizing social networks (IEEE, New York, 2009), pp. 173–187
5. M. Srivatsa, M. Hicks, in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*. De-anonymizing mobility traces: using social network as a side-channel (ACM, New York, 2012), pp. 628–637
6. X. Ying, X. Wu, in *the SIAM International Conference on Data Mining (SDM)*. Randomizing social networks: a spectrum preserving approach (Society for Industrial and Applied Mathematics, Philadelphia, 2008), pp. 739–750
7. B. Zhou, J. Pei, in *the IEEE 24th International Conference on Data Engineering (ICDE)*. Preserving privacy in social networks against neighborhood attack (IEEE, New York, 2008), pp. 506–515
8. K. Liu, E. Terzi, in *2008 ACM SIGMOD International Conference on Management of Data*. Towards identity anonymization on graphs (ACM, New York, 2008), pp. 93–106
9. L. Zou, L. Chen, M. T. Özsu, in *the VLDB Endowment*. K-automorphism: a general framework for privacy preserving network publication, (2009), pp. 946–957
10. M. Hay, G. Miklau, D. Jensen, D. Towsley, P. Weis, in *the VLDB Endowment*. Resisting structural re-identification in anonymized social networks (ACM, New York, 2008), pp. 102–114
11. L. Backstrom, C. Dwork, J. Kleinberg, in *the 16th International Conference on World Wide Web*. Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography (ACM, New York, 2007), pp. 181–190
12. S. Bhagat, G. Cormode, B. Krishnamurthy, D. Srivastava, in *the VLDB Endowment*. Class-based graph anonymization for social network data (ACM, New York, 2009), pp. 766–777
13. B. Thompson, D. Yao, in *the 4th International Symposium on Information, Computer, and Communications Security (ASIACCS)*. The union-split algorithm and cluster-based anonymization of social networks, (2009), pp. 218–227
14. A. Sala, X. Zhao, C. Wilson, H. Zheng, B. Zhao, in *the 2011 ACM SIGCOMM Conference on Internet Measurement Conference (IMC)*. Sharing graphs using differentially private graph models (ACM, New York, 2011), pp. 81–98
15. D. Proserpio, S. Goldberg, F. McSherry, in *the 2012 ACM Workshop on Workshop on Online Social Networks*. A workflow for differentially-private graph synthesis (ACM, New York, 2012), pp. 13–18
16. D. Proserpio, S. Goldberg, F. McSherry, in *the 40th International Conference on Very Large Data Base (VLDB)*. Calibrating data to sensitivity in private data analysis (ACM, New York, 2014), pp. 673–648
17. Q. Xiao, R. Chen, K. Tan, in *the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. Differentially private network data release via structural inference (ACM, New York, 2014), pp. 911–920
18. P. Mittal, C. Papamanthou, D. Song, in *the 20th Annual Network and Distributed System Security Symposium (NDSS)*. Preserving link privacy in social network based systems (arXiv, Ithaca, 2013), pp. 1–15
19. S. Nilizadeh, A. Kapadia, Y.-Y. Ahn, in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. Community-enhanced de-anonymization of online social networks (ACM, New York, 2014), pp. 537–548
20. L. Yartseva, M. Grossglauer, in *the First ACM Conference on Online Social Networks*. On the performance of percolation graph matching (ACM, New York, 2013), pp. 119–130
21. W. L. S. Ji, M. Srivatsa, J. He, R. Beyah, in *the Information Security (ISC)*. Structure based data de-anonymization of social networks and mobility traces (Springer, New York, 2014), pp. 237–254
22. N. Korula, S. Lattanzi, in *the VLDB Endowment*. An efficient reconciliation algorithm for social networks (ACM, New York, 2014), pp. 377–388
23. P. Pedarsani, D. R. Figueiredo, M. Grossglauer, in *the Allerton Conference on Communication, Control, and Computing*. A Bayesian method for matching two similar graphs without seeds (IEEE, New York, 2013), pp. 1598–1607
24. A. Korolova, R. Motwani, S. Nabar, Y. Xu, in *the 17th ACM Conference on Information and Knowledge Management (CIKM)*. Link privacy in social networks (ACM, New York, 2008), pp. 239–298
25. G. Wondracek, T. Holz, E. Kirda, C. Kruegel, in *the 2010 IEEE Symposium on Security and Privacy*. A practical attack to de-anonymize social network users, (2010), pp. 223–238
26. S. Ji, P. Mittal, R. Beyah, Graph data anonymization, de-anonymization attacks, and de-anonymizability quantification: a survey. *IEEE Commun. Surv. Tutor.* **19**(2), 1305–1326 (2016)
27. S. Ji, W. Li, N. Z. Gong, P. Mittal, R. Beyah, Seed-based de-anonymizability quantification of social networks. *IEEE Trans. Inf. Forensics Secur.* **11**(7), 1398–1411 (2016)
28. S. Ji, T. Wang, J. Chen, W. Li, P. Mittal, R. Beyah, De-sag: on the de-anonymization of structure-attribute graph data. *IEEE Trans. Dependable Secure Comput.* **PP**(99), 1–14 (2017)
29. T. Lou, J. Tang, J. Hopcroft, Z. Fang, X. Ding, Learning to predict reciprocity and triadic closure in social networks. *ACM Trans. Knowl. Discov. Data (TKDD)*. **7**(2), 5 (2013)
30. J. Tang, J. Sun, C. Wang, Z. Yang, in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Social influence analysis in large-scale networks (ACM, New York, 2009), pp. 807–816
31. X. Ying, X. Wu, in *Proceedings of the 2008 SIAM International Conference on Data Mining*. Randomizing social networks: a spectrum preserving approach (Society for Industrial and Applied Mathematics, Philadelphia, 2008), pp. 739–750

32. K. Liu, E. Terzi, in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. Towards identity anonymization on graphs (ACM, New York, 2008), pp. 93–106
33. J. Cheng, A. Fu, J. Liu, in *the 2010 ACM SIGMOD International Conference on Management of Data*. K-isomorphism: privacy preserving network publication against structural attacks (ACM, New York, 2010), pp. 459–470
34. Y. Wang, X. Wu, Preserving differential privacy in degree-correlation based graph generation[J]. *Trans. Data. Priv. (NIH Public Access)*. **6**(2), 127–145 (2013)
35. L. Backstrom, C. Dwork, J. Kleinberg, in *Proceedings of the 16th International Conference on World Wide Web*. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography (ACM, New York, 2007), pp. 181–190
36. A. Narayanan, E. Shi, B. I. Rubinstein, in *Neural Networks (IJCNN), The 2011 International Joint Conference On*. Link prediction by de-anonymization: how we won the Kaggle social network challenge (IEEE, New York, 2011), pp. 1825–1834
37. S. Ji, W. Li, M. Srivatsa, J. S. He, R. Beyah, General graph data de-anonymization: from mobility traces to social networks. *ACM Trans. Inf. Syst. Secur. (TISSEC)*. **18**(4), 12 (2016)
38. S. Ji, W. Li, M. Srivatsa, R. Beyah, in *Proceedings of ACM Sigsac Conference on Computer and Communications Security*. Structural data de-anonymization: quantification, practice, and implications (ACM, New York, 2014), pp. 1040–1053
39. K. Sharad, Learning to de-anonymize social networks (2016). Technical report, University of Cambridge, Computer Laboratory
40. W.-H. Lee, C. Liu, S. Ji, P. Mittal, R. B. Lee, in *Proceedings of the 2017 on Workshop on Privacy in the Electronic Society*. Blind de-anonymization attacks using social networks (ACM, New York, 2017), pp. 1–4
41. X. Wu, Z. Hu, X. Fu, L. Fu, X. Wang, S. Lu, in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*. Social network de-anonymization with overlapping communities: analysis, algorithm and experiments (IEEE, 2018), pp. 1–9
42. D. Chen, B. Hu, S. Xie, De-anonymizing social networks. *CS224W Course Proj. Writup.*, 1–9 (2012)
43. J. Qian, X. Y. Li, C. Zhang, L. Chen, in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*. De-anonymizing social networks and inferring private attributes using knowledge graphs (IEEE, 2016), pp. 1–9
44. H. Jiang, J. Yu, C. Hu, C. Zhang, X. Cheng, SA framework based de-anonymization of social networks. *Procedia Comput. Sci.* **129**, 358–363 (2018)
45. Z. Cai, Z. He, X. Guan, Y. Li, Collective data-sanitization for preventing sensitive information inference attacks in social networks. *IEEE Trans. Dependable Secure Comput. (TDSC)*. **15**(4), 577–590 (2018)
46. N. Z. Gong, A. Talwalkar, L. Mackey, L. Huang, E. C. R. Shin, E. Stefanov, E. Shi, D. Song, Jointly predicting links and inferring attributes using a social-attribute network (san)[J]. *arXiv preprint arXiv:1112.3265* (2011). <https://arxiv.org/abs/1112.3265>. Accessed 26 July 2018
47. B. Thompson, D. Yao, in *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*. The union-split algorithm and cluster-based anonymization of social networks (ACM, 2009), pp. 218–227
48. S. Bhagat, G. Cormode, B. Krishnamurthy, D. Srivastava, Class-based graph anonymization for social network data. *Proc. VLDB Endowment*. **2**(1), 766–777 (2009)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
