This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Structure-Attribute-Based Social Network Deanonymization With Spectral Graph Partitioning

Honglu Jiang, Jiguo Yu, *Senior Member, IEEE*, Xiuzhen Cheng, *Fellow, IEEE*, Cheng Zhang, Bei Gong, and Haotian Yu

***Abstract*— Online social networks have gained tremendous popularity and have dramatically changed the way we communicate in recent years. However, the publishing of social network data raises more and more privacy concerns. To protect user privacy, social networking data are usually anonymized before being released. Nevertheless, existing anonymization techniques do not have sufficient protection effects. A large number of deanonymization attacks have arisen, and they mainly make use of either network topology or node attribute information to successfully reidentify anonymized users. In this article, we model a social network as a structure-attribute network (SAN) integrating the structural characteristics and the attribute information associated with social network users. A novel similarity measurement of social network nodes is proposed by considering the structural similarity and attribute similarity. A two-phase scheme is then designed to perform deanonymization by first dividing a social network (graph) into smaller subgraphs based on spectral graph partitioning and then applying the proposed deanonymization algorithm on each matched subgraph pair. We simulate the deanonymization attack with extensive experiments on three real-world datasets, and the experimental results demonstrate that our approach can improve the accuracy and time complexity of deanonymization compared with the state of the art.**

***Index Terms*— Deanonymization, differential privacy, *k*-anonymity, social network.**

Manuscript received June 16, 2020; revised February 15, 2021 and April 19, 2021; accepted May 18, 2021. This work was supported in part by the US NSF under grants IIS-1741279 and CNS-1704397. *(Corresponding author: Jiguo Yu.)*

Honglu Jiang is with the Department of Computer Science, The George Washington University, Washington, DC 20052 USA, and also with the Department of Computer Science, The University of Texas Rio Grande Valley, Brownsville, TX 78520 USA (e-mail: hljiang0720@gwu.edu).

Jiguo Yu is with the School of Computer Science, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China, and also with the Shandong Laboratory of Computer Networks, Jinan 250014, China (e-mail: jiguoyu@sina.com).

Xiuzhen Cheng is with the Department of Computer Science, The George Washington University, Washington, DC 20052 USA, and also with the School of Computer Science and Technology, Shandong University, Qingdao 266510, China (e-mail: xzcheng@sdu.edu.cn).

Cheng Zhang is with the Paul and Virginia Engler College of Business, West Texas A&M University, Canyon, TX 79016 USA (e-mail: zhangchengcarl@hotmail.com).

Bei Gong is with the Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: gongbei@bjut.edu.cn).

Haotian Yu is with the Department of Data Analytics, The George Washington University, Washington, DC 20052 USA (e-mail: yuxx6789@gwu.edu).

Digital Object Identifier 10.1109/TCSS.2021.3082901

NOMENCLATURE

Notations:

Notation	Description
$G = (V, E, B)$	a social network graph
$G^a = (V^a, E^a, B^a)$	an anonymized graph
$G^u = (V^u, E^u, B^u)$	an auxiliary graph
$N^a(i)$, $N^u(i)$	the neighborhood of node i in the anonymized graph and in the auxiliary graph, respectively
$\Delta^a(i)$, $\Delta^u(i)$	the degree of node i in the anonymized graph and in the auxiliary graph, respectively
σ	the mapping $V^a \rightarrow V^u$
$S_A(i, j)$	the attribute similarity between nodes i and j
$S_R(i, j)$	the structural similarity between nodes i and j
$S(i, j)$	the similarity between nodes i and j
$Sim_\sigma(G^a, G^u)$	the similarity between G^a and G^u under the mapping σ

I. INTRODUCTION

ONLINE social networks have gained tremendous popularity and have been playing important roles in our daily life. They can provide online information sharing and serve as exchange platforms for different groups with diverse capabilities/functions, which dramatically changes the ways we communicate [1]. As a reflection of real-world social life, social networking data contain a large amount of sensitive information. In many social networking sites, users are asked to fill in personal information, such as name, gender, birthday, educational background, occupation, marital status, e-mail, or even personal photographs. In addition, texts, pictures, videos, and geographical locations published by users (user-generated contents) are retained in the social network database [2]. These data are often published to third parties for services, such as data analysis, targeted advertising, recommendations, and evaluations on applications, resulting in increasing concerns on privacy violations.

To protect user privacy, social networking data are usually anonymized before being released [3], [4]. Generally speaking, data anonymization techniques can be characterized into three

2329-924X © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: George Mason University. Downloaded on January 31,2022 at 22:53:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

categories: naive ID removal, k-anonymity [5] (including l-diversity [6] and t-closeness [7]), and differential privacy [8], [9]. These techniques achieve data anonymization mainly by removing certain unique attribute information and then perturbing network structures, which aims to improve the privacy preservation level while maintaining high data utility.

However, such anonymization schemes do not have sufficient protection effects. There have been a number of deanonymization attacks targeting the released anonymized social network data to reidentify an anonymous individual [10]–[17]. Deanonymization takes as inputs an anonymized graph and an auxiliary graph and outputs as many matched node pairs in these two graphs as possible. They are either seed-based [11], [16], [18] (assuming the availability of a set of known anonymized identities and true identity mappings) or seedless [19], [20] and resort to the network structure (node degrees and/or connectivity, and so on) and/or attribute information to deanonymize an anonymized social network. By making use of a variety of background knowledge, deanonymization can have a strong attack ability.

Nevertheless, deanonymization accuracy is significantly impacted by various factors. For example, seed-based algorithms heavily depend on the quality and quantity of the seeds, and low-quality seeds could lead to poor performance of reidentification due to error propagation and accumulation. Moreover, it is hard to collect an appropriate number of high-quality seed pairs since the published network is highly anonymized. On the other hand, the published graph data contains much non-Personal Identifiable Information (non-PII) or attribute information, such as gender, age, education, and residence. Attribute information has a great impact on the deanonymization accuracy of graph data [21], but the corresponding investigation from the existing research is far from satisfactory, let alone the joint consideration of both structural and attribute information. In this article, we propose a deanonymization approach considering both local topology and node attribute values in a social network. Our contributions can be summarized as follows (a preliminary version of this article is presented in [22]).

1) Considering that the scale of a social network is usually large, we resort to spectral graph partitioning to first partition a social network into small subgraphs and then deanonymize the subgraphs in parallel. Experimental studies reveal that, when the number of partitions is small, the deanonymization accuracy can be enhanced as error accumulation is less serious, while the accuracy is decreased when the number of partitions is large enough because too many intersubgraph links are overlooked.
2) We propose a novel similarity measurement of social network nodes by considering the structural similarity and attribute similarity, so as to improve the accuracy of deanonymization. Based on this new node similarity metric, we design a graph deanonymization algorithm that relies on a single seed pair following the BFS traversal order to gradually search for the matching nodes. The design motivation of our deanonymization algorithm lies in that the neighbors of matched nodes tend to match with a high probability, thus significantly improving the deanonymization efficiency.
3) We carry out extensive experimental studies to validate the deanonymization performance on three real-world datasets, i.e., Twitter, Facebook, and Google+, and the results indicate that our approach can obtain a good tradeoff between the efficiency and accuracy of deanonymization compared to the state of the art.

The rest of this article is organized as follows. We describe the most related work in Section II and present the structure-attribute network (SAN) model, the attack model, and the formal definition of our deanonymization problem in Section III. The spectral partitioning algorithm employed in this article to partition large social networks into small subgraphs for parallel processing, the novel node similarity metric that captures both structural similarity and attribute similarity, and the deanonymization algorithm that employs the new similarity metric are all detailed in Section IV. Our experimental studies based on three real-world datasets to evaluate the performance of the deanonymization algorithm are presented in Section V, and the results validate the good performance in terms of accuracy and efficiency. Conclusions and future research are presented in Section VI.

II. Related Work

Deanonymization approaches in social networks can be deemed as either seed-based or seedless depending on whether a set of known mappings between anonymized identities and true identities is needed. Another type of classification relies on whether the node similarity measurement is based on local social graph structures. In this section, we summarize the most related work by classifying the existing deanonymization methods as either structure-based or nonstructure-based. Structure-based deanonymization makes use of the local topology information of each node, such as the node degree and neighborhood information for deanonymization, while nonstructure-based approaches employ information other than topology.

A. Structure-Based Deanonymization

Backstrom *et al.* [10] first introduced structure-based deanonymization, where the authors proposed both active attacks and passive attacks to deanonymize social network data. The basic idea of these attacks is to create a subgraph with a special link pattern to the target users, based on which the target users can be deanonymized by identifying the previously created subgraph and the link pattern from the released anonymized graph. It is obvious that these attacks are not scalable and difficult to control due to the continuous growth of social network data during the process of the data release. Narayanan and Shmatikov [11] proposed a robust and scalable seed-based deanonymization attack for large-scale directed social networks. Their algorithm consists of two processes: seed identification and propagation, to identify a set of seed mappings in the first phase and propagate the deanonymization from the seed mappings to other users in

Authorized licensed use limited to: George Mason University. Downloaded on January 31,2022 at 22:53:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

the anonymized graph by adopting several deanonymization heuristics in the second phase.

Srivatsa and Hicks [15] designed a deanonymization attack to mobility traces using social network data as the side-channel information. They presented three two-phase schemes to perform the deanonymization attack, but the proposed schemes suffer from scalability limitations. Ji *et al.* [18] described an adaptive seed-based deanonymization (ADA) framework for the scenario where the anonymized graph and the auxiliary graph partially overlap. Ji *et al.* [23] presented a practical single-phase cold start optimization-based deanonymization algorithm. Nilizadeh *et al.* [17] developed a community-based deanonymization scheme for social networks, which can be employed to enhance seed-based attacks. Ji *et al.* [16] conducted the first perfect deanonymizability and partial deanonymizability study with seed information in general scenarios, in which social networks are assumed to follow an arbitrary network model not only ER model.

Fang *et al.* [24] proposed a structure-based weighted neighborhood matching algorithm by considering local structural features and closeness centrality of nodes when calculating node similarities. This algorithm adopts a dynamic similarity matrix and the weighted neighborhood matching to ensure good noise tolerance and deanonymization accuracy. Shao *et al.* [19] developed a fast and effective seedless deanonymization approach relying on structural information equipped with a pairwise node similarity measure. Hu *et al.* [25] considered the deanonymization of partially overlapping networks with the aid of seed nodes and provided general forms of theoretical results under the ER model.

Zhang *et al.* [26] probed into the seedless deanonymization problem, formalized it in the context of multihop adjacency relationship, and further generated the collective-form deanonymization problem, which aims to minimize the total differences between multihop adjacency matrices of the two observed networks called collective adjacency disagreement (CAD).

Xian *et al.* [27] proposed a multiview low-rank coding (MVLRC)-based deanonymization framework, in which the auxiliary network can be incorporated naturally with the target network for anonymized links' inference.

B. Nonstructure-Based Deanonymization

Qian *et al.* [20] introduced a knowledge graph to explicitly represent the prior information of an attacker for any individual user. Based on the defined knowledge graph, they formulated the process of deanonymization and privacy inference. Ji *et al.* [21] studied the impact of non-PII on the privacy of graph data with attribute information. They analyzed the attribute-based anonymity for structure-attribute graph data. The difference between this work and ours lies in that it focuses on the analysis of the impact of attribute information on data privacy, while our work defines a new similarity metric integrating the structural similarity and attribute similarity for node deanonymization.

Wang *et al.* [28] proposed a profile matching method based on the generation of user features. In [29], a deanonymization strategy was proposed, which is operated based on the information threshold. The attacker can query the selected anonymous user's attributes sequentially and can calculate the amount of information. The performance for social networks with a fixed finite number of users and for asymptotically large social networks was analyzed. Zhang *et al.* [30] explored the impact of user attributes in social network deanonymization. They first quantified the significance of attributes in a social network and then designed an algorithm by exploiting attribute-based similarity to deanonymize the social network data.

In light of the above analysis, one can see that the following aspects distinguish our work from the existing ones. First, we consider both the structure characteristics and the user's attribute information to define a novel node similarity metric for good deanonymization performance. Second, we present a deanonymization algorithm for partially overlapping networks starting from only a single seed pair and achieve scalability via spectral graph partitioning.

III. Definition and Model

In this section, we introduce our SAN model, the attack model, and the formal definition of the deanonymization problem. To make this article more readable, we summarize the notations and their semantic meanings in Nomenclature.

A. Social-Attribute Network Model

We model a social network as an undirected graph $G = (V, E)$, where V represents the users (nodes) and E represents the social relationships between the nodes in V. In addition to social relationships, each node is associated with a set of attributes. For instance, in SINA microblog, nodes are the SINA microblog users, and edges represent the friendship between different users; node attributes, such as age, gender, major, occupation, and residence, can be extracted from the user profiles.

We need to distinguish between attributes and attribute values. Each user has a finite number of attributes, such as residence, major, and occupation, and each attribute has a finite number of attribute values. For example, a user's occupation can be a doctor, an engineer, or a teacher. Let d be the total number of distinct attribute values in a social network. Then, one can employ a d-dimensional binary vector to represent the existence of the attribute values for each node. More specifically, let $\vec{b}_u$ be the attribute vector for node u. Then, an entry in $\vec{b}_u$ equal to 1 indicates that u has the corresponding attribute value and 0 otherwise. The attribute values of all social nodes are represented by matrix $B = [\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n]$, of which n is the total number of social nodes.

In this article, we use a structure-attribute network $G = (V, E, B)$ to model a social network consisting of a social structure $G = (V, E)$ and an attribute matrix B. Assume that an attacker has the certain background knowledge to help him complete the deanonymization. Before attacking, the adversary holds two graphs: an anonymized graph $G^a = (V^a, E^a, B^a)$ published by the OSN service provider and an auxiliary graph $G^u = (V^u, E^u, B^u)$ constructed by the adversary based on

Authorized licensed use limited to: George Mason University. Downloaded on January 31,2022 at 22:53:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

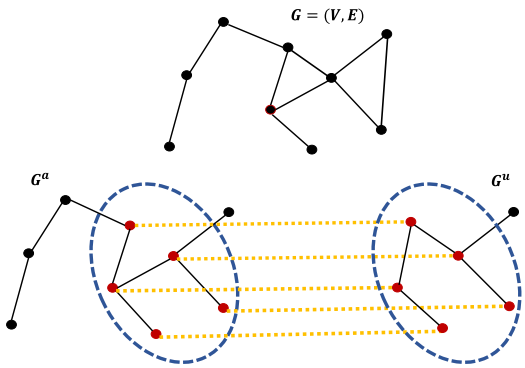


Fig. 1. Example of deanonymization.

his background knowledge. Note that the auxiliary graph is partially or completely overlapping with the anonymized graph. Given $i \in V^a$, its neighborhood is defined as $N^a(i) = \{v \in V^a | e^a_{i,v} \in E^a\}$, and the degree of node i is denoted as $\Delta^a(i) = |N^a(i)|$. Similarly, $N^u(j)$ and $\Delta^u(j)$ denote the neighborhood and the degree of node j in the auxiliary graph. For the sake of simplicity, we use $G = (V, E)$ to represent $G = (V, E, B)$ if clear from the context.

In practice, social networking data are usually anonymized to generate an anonymized graph before it is released, while the auxiliary graph can be obtained based on a variety of ways, such as data mining, cooperative information systems, and knowledge/data attacks [16]. As shown in Fig. 1, graph $G = (V, E)$ represents the original graph maintained by the OSN service provider, while graphs G^a and G^u are extracted from G with overlapping nodes and edges, and five nodes in G^a are mapped to five nodes in G^u.

B. Attack Model

The goal of deanonymization is to map the nodes in G^a to the nodes in G^u as accurately as possible. Given G^a and G^u, we can employ a mapping to formally define a deanonymization attack: $\sigma : V^a \to V^u$. $\forall i \in V^a$, its mapping under σ is $\sigma(i) \in V^u \bigcup \{\perp\}$, where $\perp$ is a special not existing indicator. Under σ, a successful deanonymization 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 $\sigma(i) = \perp$ if $\sigma(i) \notin V^u$. Otherwise, the attack on i fails. Accordingly, our goal of a deanonymization attack is to successfully deanonymize as many users in V^a as possible.

C. Problem Definition

As mentioned earlier, a deanonymization scheme can be defined as a mapping: $\sigma = V^a \to V^u$, which maximizes the total number of matched node pairs in G^a and G^u. To formally define our problem, we need a parameter "similarity" to quantify the probability of i in G^a being mapped to j in G^u, and our goal is to find a mapping σ that maximizes the total similarity of the nodes in G^a and their corresponding mapping nodes in G^u. We use Sim to measure the node similarity between G^a and G^u after matching by σ, that is,

$$\underset{\sigma}{\mathrm{Sim}}(G^a, G^u) = \sum_{(i,\sigma(i)=j)} S(i, j) \tag{1}$$

of which $S(i, j)$ denotes the node similarity between $i \in V^a$ and $j \in V^u$ and is defined in Section IV-B. As a result, our deanonymization problem can be formally stated as follows.

Definition 1 *(Deanonymization Problem):*

Input: An anonymized graph G^a and an auxiliary graph G^u.

output: A mapping σ.

Goal: maximizing the similarity between G^a and G^u, i.e., $\mathrm{Sim}_\sigma(G^a, G^u)$.

IV. Deanonymization

In this section, we first introduce a graph partitioning algorithm [31] to divide a large-scale social graph into smaller subgraphs. Then, we formally define the similarity measurement of two nodes with the consideration of structural characteristics and attributes of a social network before proposing the deanonymization algorithm.

A. Spectral Graph Partitioning

For large-scale social networks, the computation cost of deanonymization is high, and the accuracy is low. An effective method is to partition a large social network into small subgraphs. Graph partitioning can also effectively reduce the error accumulation of deanonymization. In this section, we briefly introduce the graph partitioning algorithm proposed in [31], which can achieve a significantly higher partitioning quality for social networks than other schemes [31].

We first introduce the simple case of bisection spectral graph partitioning, which produces two subgraphs with certain properties. Let $S \subseteq V$ be a set of vertices; then, its boundary is a set of edges $\partial(S) \subset E$ with each having only one endpoint in S. In other words

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

The bisection spectral graph partitioning intends to find a minimum balanced cut $C = (S, \bar{S})$ of graph $G = (V, E)$ in the sense that S satisfies either

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

or

$$\eta(S) = \min_S \frac{|\partial(S)|}{\sum_{i \in S} \Delta(i) \cdot \sum_{j \in \bar{S}} \Delta(j)} \tag{4}$$

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

Next, we generalize bisection to multiple partitions. For a graph $G = (V, E)$, let matrix $A = [a_{i,j}]$ be its weighted adjacency matrix. Notice that G is an unweighted graph under our consideration, in which edge weights can be considered

Authorized licensed use limited to: George Mason University. Downloaded on January 31,2022 at 22:53:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

to be all one. Thus, the matrix A of graph G can be defined as follows:

$$a_{i,j} = \begin{cases} 1, & \text{if } e_{i,j} \in E \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Given the adjacent matrix A, its diagonal matrix is $D = \text{diag}(Ae)$, where the vector $e = (1, \ldots, 1)^{\text{T}}$. For an unweighted graph G, the entry $a_{i,j} = 1$ denotes the existence of an edge between nodes i and j, and the diagonal of matrix D denotes the degrees of the nodes. The Laplace matrix L is defined as $L = D - A$. Since D is a diagonal matrix, one can find that $Le = 0$.

Let t be the number of subgraphs produced by graph partitioning. Denote by S_p the set of nodes in one partition, of which $p = 1, \ldots, t$. The matrix $U = [u_{i,p}]$ denotes a set of vectors $U = [\vec{u_1}, \ldots, \vec{u_t}]$, where each vector $\vec{u_p}$ corresponds to the set S_p with elements

$$u_{i,p} = \begin{cases} \theta, & \text{if } i \in S_p \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

of which θ is a constant number depending on the kind of cut under consideration.

For the ratio cut, we have $\theta = (1/(|S_p|)^{1/2})$, and

$$\vec{u_p}^{\text{T}} L \vec{u_p} = \frac{|\partial(S_p)|}{|S_p|} \tag{7}$$

$$U^{\text{T}} U = I. \tag{8}$$

Then, the graph partitioning problem can be reformulated as

$$\tilde{\rho}(S_1, \ldots, S_t) = \min_{S_1, \ldots, S_t} \sum_{p=1}^{t} \frac{|\partial(S_p)|}{|S_p|} = \min_{S_1, \ldots, S_t} \sum_{p=1}^{t} \vec{u_p}^{\text{T}} L \vec{u_p}. \tag{9}$$

The solution to the optimization problem (9) is the eigenvector associated with the t smallest eigenvalues of the following problem:

$$LU = U\Sigma. \tag{10}$$

For the normalized cut, we have $\theta = (1/(\sum_{i \in S_p} \Delta(i))^{1/2})$, and

$$\vec{u_p}^{\text{T}} L \vec{u_p} = \frac{|\partial(S_p)|}{\sum_{i \in S_p} \Delta(i)} \tag{11}$$

$$U^{\text{T}} D U = I \tag{12}$$

where I is the $t \times t$ identity matrix. Then, the graph partitioning problem can be reformulated as

$$\tilde{\eta}(S_1, \ldots, S_t) = \min_{S_1, \ldots, S_t} \sum_{p=1}^{t} \frac{|\partial(S_p)|}{\sum_{i \in S_p} \Delta(i)} = \min_{S_1, \ldots, S_t} \sum_{p=1}^{t} \vec{u_p}^{\text{T}} L \vec{u_p}. \tag{13}$$

The solution to the optimization problem (13) is the eigenvector associated with the t smallest eigenvalues of the following problem:

$$LU = DU\Sigma \tag{14}$$

where $\Sigma = \text{diag}([\lambda_1, \ldots, \lambda_t])$ and λ_t is the eigenvalue of L.

Algorithm 1 Spectral Graph Partitioning

1: Let $G = (V, E)$ be an input graph, A be its adjacent matrix, and D be its Diagonal matrix;
2: Compute the Laplacian matrix $L = D - A$;
3: Let $B = I$ for a ratio cut or $B = D$ for a normalized cut; let t be the number of partitions;
4: Find the t smallest eigenpairs of the eigenvalue problem $LU = BU\Sigma$;
5: Run a clustering algorithm such as k-means on the points defined by the rows of U.

After defining the two different partitions, we outline the graph partitioning algorithm in Algorithm 1 [31].

By applying Algorithm 1, one can partition a large social graph into t small subgraphs. Both the anonymized graph and the auxiliary graph are divided into subgraphs by the same approach with similar adjacency matrix and Laplace matrix as, in reality, the noise ratio for anonymization is not high. Then, the corresponding subgraphs can be matched according to the t largest eigenvalues. Denote by $(G_1^a, G_1^u), (G_2^a, G_2^u), \ldots, (G_t^a, G_t^u)$ the t matched subgraph pairs.

B. Similarity Measurement

In this section, we formally define the similarity of two nodes. The similarity $S(i, j)$ between nodes i and j considers the attribute similarity and the structural similarity.

1) Attribute Similarity: The attribute similarity of node $i \in V^a$ and $j \in V^u$ is denoted as $S_A(i, j)$. Let $\vec{b_i^a}$ and $\vec{b_j^u}$ be

$$\vec{b_i^a} = \left(b_{1i}^a, b_{2i}^a, b_{3i}^a, \ldots, b_{di}^a\right) \tag{15}$$

$$\vec{b_j^u} = \left(b_{1j}^u, b_{2j}^u, b_{3j}^u, \ldots, b_{dj}^u\right) \tag{16}$$

of which d is the number of attribute values. We have

$$S_A(i, j) = \frac{\vec{b_i^a} \bullet \vec{b_j^u}}{\sum_{x=1}^{d} \left(b_{xi}^a \bigoplus b_{xj}^u\right) + \vec{b_i^a} \bullet \vec{b_j^u}}. \tag{17}$$

Note that the value of $S_A(i, j)$ is between 0 and 1. If two nodes have a greater number of the same attribute values, their attribute similarity is higher.

2) Structural Similarity: Let $S_R(i, j)$ be the structural similarity between $i \in V^a$ and $j \in V^u$. The degrees of i and j are, respectively, $\Delta^a(i)$ and $\Delta^u(j)$, which are defined in Section III.

We first consider the degrees of the 1-hop neighbors of nodes i and j. Sort the degrees of the 1-hop neighbors of nodes i and j in descending order. Let $\alpha = \min(\Delta^a(i), \Delta^u(j))$; then, we extract the largest α degrees of the 1-hop neighbors of nodes i and j and construct two α-dimensional vectors $\vec{D_\alpha^a}(i)$ and $\vec{D_\alpha^u}(j)$.

Similarly, we consider the degrees of the two-hop neighbors of nodes i and j. Sort the degrees of the two-hop neighbors $N_2^a(i)$ and $N_2^u(j)$ of nodes i and j in descending order. Let $\beta = \min(|N_2^a(i)|, |N_2^u(j)|)$; then, we extract the largest

Authorized licensed use limited to: George Mason University. Downloaded on January 31,2022 at 22:53:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

β degrees of the two-hop neighbors of nodes i and j and construct two β-dimensional vectors $D^{\vec{a}}_{\beta}(i)$ and $D^{\vec{u}}_{\beta}(j)$.

Then, we combine the two vectors of $D^{\vec{a}}_{\alpha}(i)$ and $D^{\vec{a}}_{\beta}(i)$ to get $[D^{\vec{a}}_{\alpha}(i), D^{\vec{a}}_{\beta}(i)]$; similarly, we get $[D^{\vec{u}}_{\alpha}(j), D^{\vec{u}}_{\beta}(j)]$. Next, we compute the cosine similarity of the two combination vectors, of which the value of parameter $0 \leq c_1 \leq 1$ is determined by experimental studies. Cosine similarity is a measure of similarity between two nonzero vectors, which is the most commonly used in high-dimensional positive spaces

$$S_R(i,j) = c_1\left(1 - \frac{|\Delta^a_i - \Delta^u_j|}{\max\left\{\Delta^a_i, \Delta^u_j\right\}}\right) + (1-c_1)\cos\left(\left[D^{\vec{a}}_{\alpha}(i), D^{\vec{a}}_{\beta}(i)\right], \left[D^{\vec{u}}_{\alpha}(j), D^{\vec{u}}_{\beta}(j)\right]\right). \tag{18}$$

3) Node Similarity: The node similarity between $i \in V^a$ and $j \in V^u$ is denoted as $S(i,j)$, which is computed as

$$S(i,j) = c_2 S_A(i,j) + (1-c_2) S_R(i,j) \tag{19}$$

of which c_2 is a weight coefficient with $0 \leq c_2 \leq 1$. If $c_2 > 0.5$, it means that the attribute information is more important; otherwise, structural similarity is weighed higher.

C. Deanonymization Algorithm

We employ (G^a_s, G^u_s) to represent a matched subgraph pair after graph partitioning. The objective of our deanonymization attack is to find a mapping that maximizes the similarity of the social nodes in G^u_s and those in G^a_s. Intuitively, one can construct a weighted complete bipartite graph $G^B_s = (V^a_s + V^u_s, \varepsilon^B_s)$, of which ε^B_s is the set of edges between the nodes in V^a_s and those in V^u_s. The similarity value $S(i,j)$ is assigned to link $e_{i,j} \in \varepsilon^B_s$ as its weight. Then, the deanonymization problem is reduced to the maximum weighted bipartite matching problem, which can be solved by the Hungarian algorithm. However, constructing the complete bipartite graph has large time and space complexities as we have to calculate the similarity between any pair of nodes in G^u_s and G^a_s. To address this problem, we build a lightly weighted bipartite graph in Algorithm 2. The basic idea is to find the best possible k candidate matching nodes from V^u_s for each node in V^a_s, starting from an initial node pair following the BFS traversal order.

Algorithm 2 takes as inputs the matched subgraph pair (G^a_s, G^u_s), an initial node pair (p^a_0, p^u_0), and parameters k and r, where r is the similarity threshold for selecting the k candidate nodes, and outputs a maximum weighted bipartite matching σ. The initial node $p^a_0 \in V^a_s$ and its matching node $p^u_0 \in V^u_s$ can be selected by randomly choosing a node in V^a_s and calculate its similarities with all the nodes in V^u_s to examine if there is a successful mapping, i.e., the largest similarity is greater than a threshold and is evidently greater than the second largest similarity. As the correctness of the initial node pair strongly affects the attack performance, we might repeat this step a few times to get the right initial node pair with high confidence.

Starting from p^a_0 and its matching node p^u_0, we traverse the nodes in V^a_s based on the BFS ordering. For any node $i \in V^a_s$, its parent node in the BFS is denoted as $p(i)$, and its candidate node set in V^u_s is denoted by Can_i. Candidate nodes are selected based on the observation that, if two nodes match, their neighbors are likely to match. We first initialize G^B_s to be a bipartite graph with node set $V^a_s \cup V^u_s$ and an empty edge set (line 1). Then, $\text{Can}_{p^a_0}$ is initialized to include only p^u_0 (line 2). Next, for each node $i \in V^a_s$, the algorithm searches $p(i)$'s candidate nodes and their neighbors in G^u_s, computes the similarities, and then compares the similarities with the threshold r to obtain at most k candidate matching nodes with the largest similarities to i (lines 3–17). After obtaining Can_i for each node $i \in V^a_s$, we construct a link from i to each node in Can_i and insert it into ε^B_s to get the bipartite graph G^B_s (lines 18–21).

Note that, in our algorithm, the initial node p^a_0 needs to compare with all the nodes in V^u_s to find out its matching node p^u_0. Other nodes only need to compare with the neighbors of their parent nodes' candidate matching nodes, which can greatly decrease the time complexity. Specifically, 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 $O(n^a_s \cdot n^u_s)$ (the number of social nodes in the matched subgraph pair G^a_s and G^u_s). 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_s is linked to top-k candidate nodes in V^u_s. Accordingly, the number of links is reduced to $O(k \cdot n^a_s)$. Thus, the time complexity of solving the maximum weighted bipartite matching problem is lowered.

Also, note that the predefined parameters k and r can trade off the deanonymization accuracy and algorithm efficiency: when k is too large or r is too small, the bipartite graph may have too many unnecessary edges for mapping; conversely, the matching process may miss important links. We will evaluate the impact of these two parameters through experimental studies.

V. Experimental Evaluations

In this section, we study the performance of our deanonymization algorithm and compare it with the state-of-the-art approaches on three real datasets of Twitter, Facebook, and Google+.

A. Datasets' Descriptions

1) Twitter Dataset: We collected the social structures (including social users and their relationships) and user attributes from Twitter using the Twitter API. Then, we constructed an undirected social network with an undirected link between user i and user j if i is in j's friend list or j is in i's friend list.

The attribute data that we crawled from Twitter include "user id," "screen name," "user name," "create time," "city," "time zone," and "biography." We considered two attributes, i.e., "create time" and "city," in our experimental study. It is noted that the users can fill in their profiles freely resulting in many infrequent or meaningless attribute values. Moreover,

Authorized licensed use limited to: George Mason University. Downloaded on January 31,2022 at 22:53:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Algorithm 2 SAN-Based Deanonymization (SAN-DA)

Input: a matched subgraph pair $(G_s^a = (V_s^a, E_s^a), G_s^u = (V_s^u, E_s^u))$, an initial node $p_0^a \in V_s^a$ and its mapping node $p_0^u \in V_s^u$, parameters k and r.

Output: a maximum weighted bipartite matching σ

```
Define ε^B = ∅, build a bipartite graph G_s^B = (V_s^a ∪ V_s^u, ε_s^B)
Can_{p_0^a} = p_0^u
for each i ∈ V_s^a following the BFS order starting from p_0 do
  Define Can_i = ∅
  for each v ∈ Can_{p(i)} do
    // p(i) denotes the parent node of i.
    if S(i, v) > r then
      Can_i = Can_i ⋃ {v}
    end if
    for each neighbor j of v in G_s^u do
      if S(i, j) > r then
        Can_i = Can_i ⋃ {j}
      end if
    end for
  end for
  Update Can_i to the k nodes with the first k highest similarities
end for
for each i ∈ V_s^a do
  for each a ∈ Can_i do
    ε_s^B = ε_s^B ⋃ e_{i,a}
  end for
end for
Execute the Hungarian algorithm on the bipartite graph G_s^B
Return a maximum weighted bipartite matching σ
```

typos of inputs sometimes make the same attribute value different. Thus, we preprocessed the data and removed the incomplete, invalid, or duplicate attribute values. Also, note that the Twitter website records the "create time" of each user, but we only quoted "year" as the attribute in this study. We selected as attribute values the top nine years for "create time" and the top-70 cities for "city" in which most users claimed that they have lived.

Finally, we obtained a dataset consisting of 7910 users, 874 222 undirected social links, and 79 distinct attribute values.

2) Facebook and Google+ Datasets: The datasets of Facebook and Google+ were obtained from the Stanford Network Analysis Project (SNAP) [32].

The Facebook dataset contains "education school," "hometown," and some other anonymized features. We processed the network data by removing the invalid and duplicate attribute values. Moreover, we added gender as a new user attribute by randomly assigning a gender value to each social network user. After this processing, we obtained a Facebook dataset, including three attributes: "education school," "hometown," and "gender." This dataset contains 4032 nodes, 88 234 undirected social links, and 112 distinct attribute values, including 40 education schools, 70 hometowns, and two genders.

TABLE I
BASIC STATISTICS OF THE SANS

Dataset	Social nodes	Edges	Attribute values	Average Degree
Twitter	7910	874222	79	210.9
Facebook	4032	88234	112	43.69
Google+	3859	4992	67	2.59

The Google+ dataset contains the "education school," "major," "province," "city," and "gender." We chose three attributes of "province," "city," and "gender." This dataset contains 3859 nodes, 4992 social links, and 67 distinct attribute values, including 35 provinces, 30 cities, and two genders.

B. Constructing an SAN

We took each social user as a node and the friendships between users as edges to construct a network. For each dataset, we constructed the attribute matrix B to represent the attribute information associated with the users. Table I shows the basic statistics of our constructed SANs.

1) Anonymized Graph: We first generated an anonymized graph before conducting deanonymization. For this purpose, we employed the parameter p to denote the degree of anonymization. The larger the p, the greater number of edges that are modified. The anonymization algorithms used in our experiments include the following.

1) *Naive Anonymization:* The naive approach simply substitutes the user IDs with random anonymous identifiers, while the graph structure remains unchanged.
2) *Switch Anonymization:* The switch approach randomly selects two edges (i_1, j_1) and (i_2, j_2) and exchanges their endpoints, that is, (i_1, j_1) and (i_2, j_2) are deleted, and (i_1, j_2) and (i_2, j_1) are inserted to the network. If we intend to add p noise to the anonymized graph, this process needs to be repeated $(p/2) \times |E^a|$ times.
3) *Perturb Anonymization:* The perturb anonymization randomly deletes some links and then randomly inserts the same number of edges. If we aim at adding p noise to the anonymized graph, we need to randomly delete $(p/2) \times |E^a|$ links and then add the same number of random links.

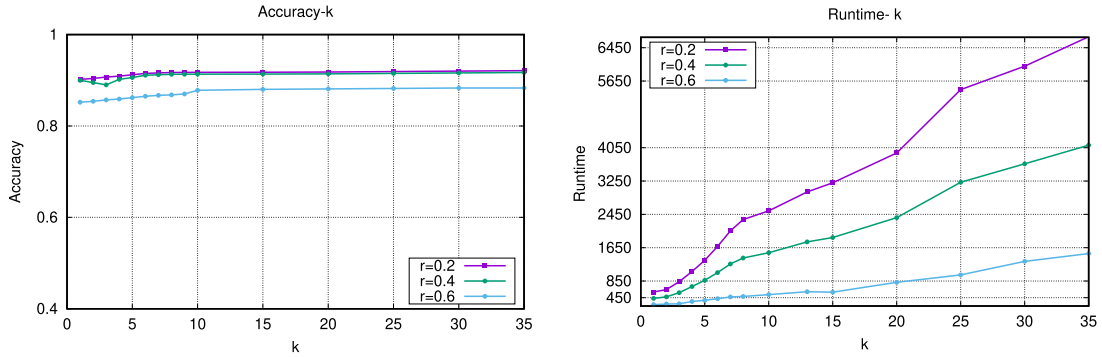
2) Anonymized Attribute Information: We also anonymized the attribute information of social network users before conducting deanonymization. For this purpose, we employed the parameter q to denote the degree of anonymization. The larger the q, the greater number of attribute values that are modified. That is, the larger the q, the greater number of elements in matrix B that is perturbed. If we aim at adding q noise to the attribute values, we need to randomly flip $q \times (n \cdot d)$ elements of B, of which n is the number of social nodes of one graph and d is the number of attribute values.

3) Auxiliary Graph: The auxiliary graph was the reduced graph of V^u, which contains 80% of the nodes randomly selected from V.

4) Graph Partitioning: Both the anonymized graph and the auxiliary graph were partitioned into t number of subgraphs based on Algorithm 1 described in Section IV-A.

Authorized licensed use limited to: George Mason University. Downloaded on January 31,2022 at 22:53:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) (b)

Fig. 2. Experimental results versus k on Twitter with $p = 0.1$ and $q = 0.01$. (a) Accuracy versus k. (b) Runtime versus k.

The corresponding subgraphs were matched according to the t eigenvalues. As a result, we obtained the following t matched subgraph pairs: $(G_1^a, G_1^u), (G_2^a, G_2^u), \ldots, (G_t^a, G_t^u)$.

C. Experimental Results

We employed *accuracy* and *run time* as the performance metrics to evaluate our deanonymization algorithm. *Accuracy* was defined as $(|M_{\text{correct}}(i, j)|)/(|V^a \cap V^u|)$, of which $M_{\text{correct}}(i, j)$ indicates the number of correctly matched node pairs. Our deanonymization algorithm was applied on the three datasets of Twitter, Facebook, and Google+ introduced in Section V-A.

1) Performance Results Versus Different Parameter Values: In this section, we evaluated the impact of parameters c_1, c_2, k, r, and t on the deanonymization performance. Recall that c_1 is a weight coefficient measuring the relative importance of one- and two-hop neighborhoods in structural similarity, while c_2 and $1 - c_2$ are the weights assigned to attribute similarity and structural similarity in determining the overall node similarity. Both c_1 and c_2 play important roles in measuring the node similarity. On the other hand, k and r are employed to tradeoff the deanonymization accuracy and the algorithm complexity, of which k is the number of candidate matching nodes for each node and r is a similarity threshold to help select the candidate nodes. The last parameter under our consideration is t, which is the number of subgraphs obtained from the graph partitioning algorithm.

We conducted our experiments on the Twitter dataset to determine these parameters since the larger the amount of data, the more accurate the parameter selection. Note that we only evaluated the results based on perturb anonymization, as the naive anonymization and switch anonymization are its special cases. We first determined c_1, c_2, k, and r without considering partitioning, based on which we then figured out the value of t. Each experiment was repeated 100 times to obtain an averaged result.

Determined after many trials, we set $c_1 = 0.4$ and $c_2 = 0.4$ since we only adopt two or three attributes for each dataset. With the aid of graph partitioning, Algorithm 2 can be parallelized in different subgraphs, so the runtime decreases with the increase in t due to parallelism, and the accuracy can be improved with the increase in t when t is less than a certain value because error accumulation becomes less serious when t gets larger. However, when t is too large, the accuracy slightly decreases as more intersubgraph structure information is not used for deanonymization. More specifically, we choose $t = 4$ for the Twitter dataset and $t = 2$ for Facebook and Google+ in the following experimental studies.

Fig. 2 illustrates the impact of parameters k and r on the accuracy and runtime of Algorithm 2 based on the Twitter dataset when the perturb noise $p = 0.1$ and $q = 0.01$. From Fig. 2(a), one can see that the accuracy can be improved with the increasing k when $k \leq 8$, and the improvement becomes not obvious when $k > 8$. Moreover, the runtime constantly increases with the growing k, as shown in Fig. 2(b). Thus, we choose $k = 8$ in the following studies. Moreover, when r increases, the obtained bipartite graph becomes sparser, and some critical links could be missed, causing the involved node matchings to fail; thus, the accuracy and runtime both greatly decrease. Therefore, we choose $r = 0.4$ in the following experiments. The simulation results shown in Figs. 3 and 4 demonstrate similar conclusions for the datasets of Facebook and Google+.

2) Comparison Studies:

a) Accuracy: We compared our algorithm with the most influential deanonymization method proposed in [11], which serves as a baseline, the ADA algorithm in [18], which employs a unified similarity (US) measurement considering both the local and global structural characteristics, and the De-SAG algorithm in [21], which also considers structural information and attribute information. We considered three different anonymization techniques, namely, naive, switch, and perturb to generate an anonymized graph. One can see that from Fig. 5 where $p = 0.1$ and $q = 0$, different deanonymization algorithms on different datasets all achieve high accuracy for the naive anonymization method since the naive approach does not change the graph structure. For the switch and perturb anonymization methods, our algorithm achieves better performance than the other two algorithms.

We also discussed the impact of the noise rate p of the anonymized graph with perturb anonymization and the one

Authorized licensed use limited to: George Mason University. Downloaded on January 31,2022 at 22:53:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

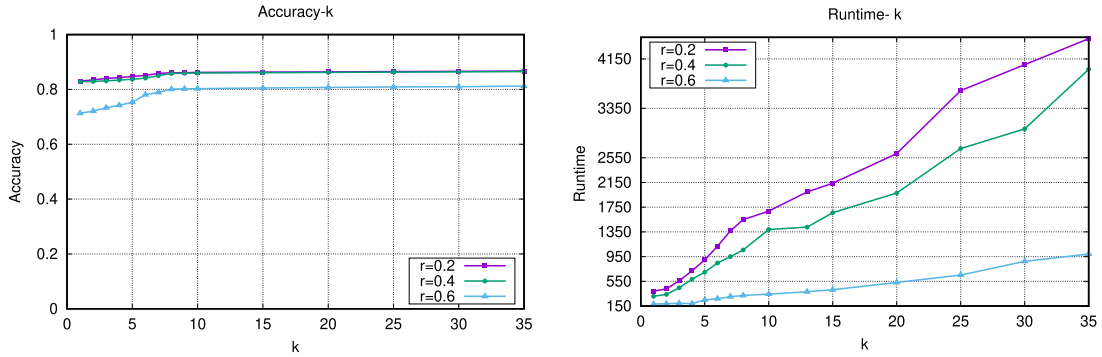


Fig. 3. Experimental results versus k on Facebook with $p = 0.1$ and $q = 0.01$. (a) Accuracy versus k. (b) Runtime versus k.

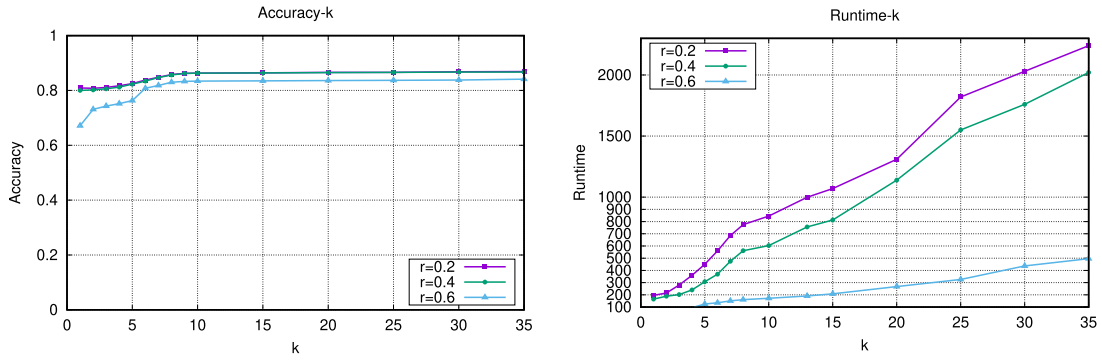


Fig. 4. Experimental results versus k on Google+ with $p = 0.1$ and $q = 0.01$. (a) Accuracy versus k. (b) Runtime versus k.

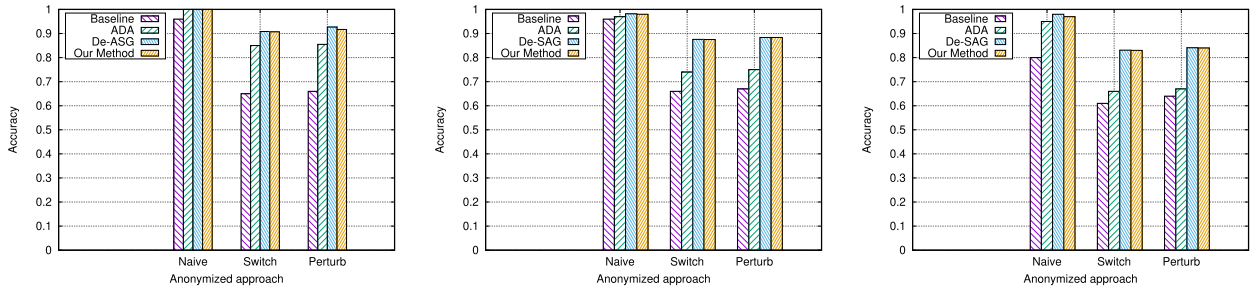


Fig. 5. Comparison results based on different anonymization methods ($p = 0.1$ and $q = 0$). (a) Twitter. (b) Facebook. (c) Google+.

of attribute information q on the matching accuracy. Fig. 6 reports the comparison results of deanonymization accuracy with different p's when $q = 0.01$ on the three datasets. We did not perturb the attribute information too much since the attribute information of these datasets is already general information, and perturbation on generalized information can greatly ruin data utility. It can be seen that our method achieves the best performance. Even, for a high noise ratio, our algorithm can still guarantee high accuracy. Our method achieves almost the same performance with De-SAG, while the structural similarity in De-SAG also considers betweenness centrality and closeness centrality. Although it can improve accuracy, the computational complexity is relatively high, while our method achieves the same performance with high efficiency, which will be illustrated in the following section. It is worth noting that, when more than 40% links are modified, the structure of the social network graph is significantly changed. Nevertheless, in reality, a publisher would not dramatically change the structure of a real dataset for data utility.

The impact of q on the performance can be demonstrated in Fig. 7. As q increases, the deanonymization accuracy of the baseline and ADA does not change since these two methods do not consider attribute information, while the results of De-SAG and our method decrease since the anonymity increases.

Note that the improvement of our method and De-SAG over the other two approaches is higher for a larger p in most cases.

Authorized licensed use limited to: George Mason University. Downloaded on January 31,2022 at 22:53:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

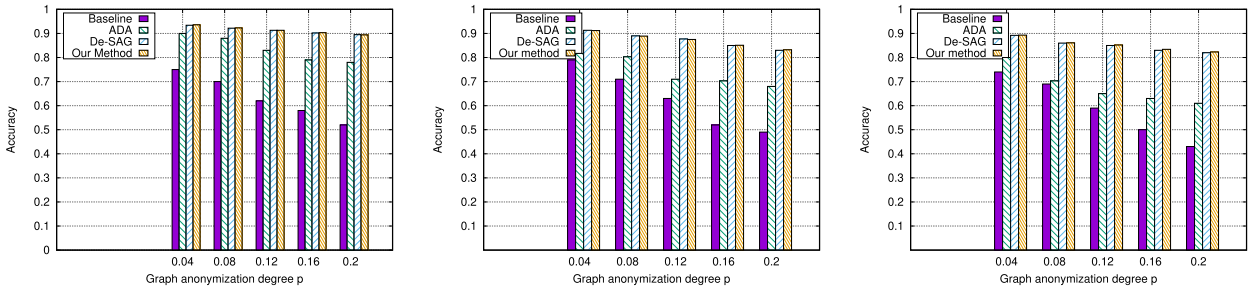


Fig. 6. Comparison results with different p's (graph anonymization method="perturbation" and $q = 0.01$). (a) Comparison result on Twitter. (b) Comparison result on Facebook. (c) Comparison result on Google+.

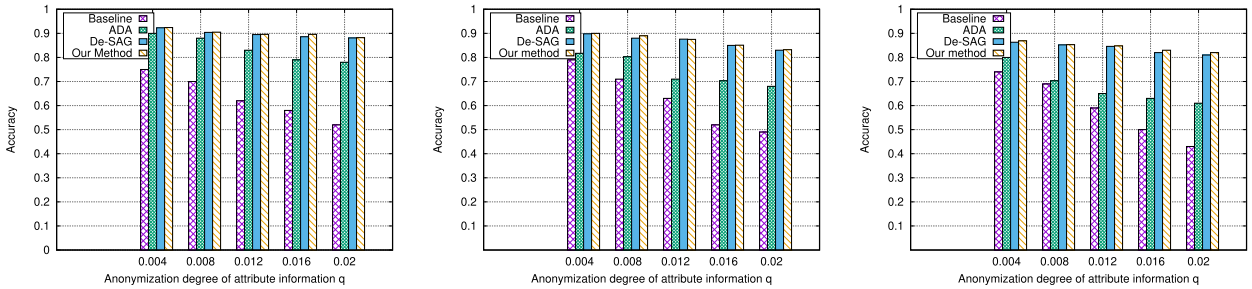


Fig. 7. Comparison results with different q's (graph anonymization method="perturbation" and $p = 0.1$). (a) Comparison result on Twitter. (b) Comparison result on Facebook. (c) Comparison result on Google+.

The reason is that the attribute information associated with users can relatively provide help for deanonymization when more structural information is perturbed.

One also notices that, for the Twitter dataset, the accuracy does not decrease too much with the increase of the perturbation ratio p, which varies from 0.936 to 0.894. However, for the datasets of Facebook and Google+, the noise ratio has an obvious effect on accuracy. More specifically, the accuracy has a great drop from 0.912 to 0.832 for the Facebook dataset and from 0.893 to 0.823 for the Google+ dataset. This is because the number of social nodes is not large, but the number of social links is very large in the Twitter dataset, resulting in a high average degree; thus, when we perturb the social graph, the change is not so obvious.

b) Runtime: As we can see from Figs. 6 and 7, our method and De-SAG achieve almost the same performance of accuracy. Based on our theoretical analysis, we improve the time complexity and deanonymization efficiency from three aspects: 1) the process of spectral graph partitioning allows deanonymizing in parallel; 2) the deanonymization algorithm reduces the mapping complexity; and 3) the metric of node similarity is time-efficient. Therefore, we explore the average runtime of our method and De-SAG on the three datasets to demonstrate our method's superiority. Fig. 8 illustrates the runtime results when the perturbation noise $p = 0, 1$, $q = 0.01$, $k = 8$, and $r = 0.4$. The runtime includes the process of spectral graph partitioning, the highest runtime of the deanonymization on all subgraphs. As shown in Fig. 8, the runtime of our algorithm is much lower than that of De-SAG, while they obtain almost the same accuracy, which demonstrates our theoretical results.

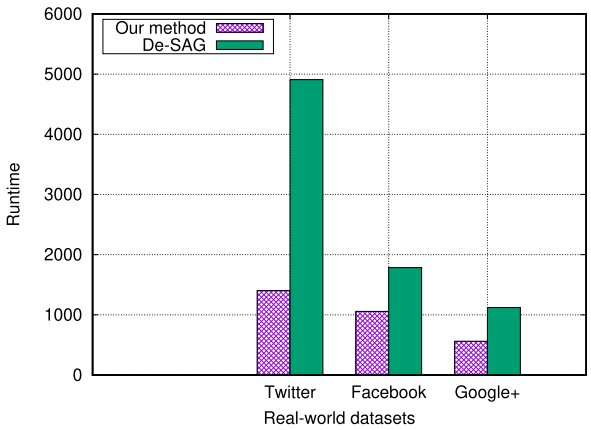


Fig. 8. Comparison results on average runtime based on different deanonymization methods.

Discussion: The good performance of our algorithm is mainly attributed to the consideration of both attribute similarity and structural similarity, which can greatly improve the matching accuracy. In our evaluations, we only used two or three attributes. Nevertheless, the greater the number of attributes adopted, the higher the accuracy. On the other hand, the time and space complexities of the algorithm increase with the adoption of a larger attribute matrix. However, a careful study indicates that the matrix is sparse. Therefore, when we analyze the matrix and calculate the node similarities, we can perform optimization techniques such as compression processing to improve the runtime and space utilization.

VI. Conclusion

Social network deanonymization is an effective approach to test the preservation level of anonymization techniques. With

Authorized licensed use limited to: George Mason University. Downloaded on January 31,2022 at 22:53:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

the results of deanonymization, one can dig out how the graph anonymization techniques affect the network properties and provide similarity analysis on different structural characteristics. In this article, we model a social network as a SAN that integrates the structural characteristics and the attribute information of the users. We propose a two-phase scheme to perform deanonymization. First, a social network (graph) is divided into smaller subgraphs by spectral graph partitioning. Subsequently, we apply a deanonymization algorithm on the matched subgraph pairs by considering both attribute similarity and structural similarity. Comprehensive experimental results on real-world datasets demonstrate that our approach can obtain a good tradeoff between the efficiency and accuracy of deanonymization. For future research, we intend to define a multimeasurement with the consideration of a greater number of similarity metrics for effective deanonymization.

REFERENCES

[1] D. M. Boyd and N. B. Ellison, "Social network sites: Definition, history, and scholarship," *J. Computer-Mediated Commun.*, vol. 13, no. 1, pp. 210–230, Oct. 2007.

[2] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: An online social network with user-defined privacy," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2009, pp. 135–146.

[3] B. Zhou, J. Pei, and W. Luk, "A brief survey on anonymization techniques for privacy preserving publishing of social network data," *ACM SIGKDD Explor. Newslett.*, vol. 10, no. 2, pp. 12–22, Dec. 2008.

[4] X. Wu, X. Ying, K. Liu, and L. Chen, "A survey of privacy-preservation of graphs and social networks," in *Managing and Mining Graph Data*. Boston, MA, USA: Springer, 2010, pp. 421–453.

[5] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, Oct. 2002.

[6] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: Privacy beyond *k*-anonymity," in *Proc. 22nd Int. Conf. Data Eng. (ICDE)*, Mar. 2006, p. 24.

[7] N. Li, T. Li, and S. Venkatasubramanian, "*t*-closeness: Privacy beyond *k*-anonymity and *l*-diversity," in *Proc. IEEE 23rd Int. Conf. Data Eng.*, Apr. 2007, pp. 106–115.

[8] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds. Springer, 2006, pp. 1–12.

[9] N. Li, W. Qardaji, D. Su, Y. Wu, and W. Yang, "Membership privacy: A unifying framework for privacy definitions," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2013, pp. 889–900.

[10] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography," in *Proc. 16th Int. Conf. World Wide Web*, 2007, pp. 181–190.

[11] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," 2009, *arXiv:0903.3276*. [Online]. Available: http://arxiv.org/abs/0903.3276

[12] A. Narayanan and V. Shmatikov, *Robust de-Anonymization of Large Datasets (how to Break Anonymity of the Netflix Prize Dataset)*. Austin, TX, USA: Univ. Texas at Austin, 2008.

[13] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel, "A practical attack to de-anonymize social network users," in *Proc. IEEE Symp. Secur. Privacy*, May 2010, pp. 223–238.

[14] M. Korayem and D. Crandall, "De-anonymizing users across heterogeneous social computing platforms," in *Proc. 7th Int. AAAI Conf. Weblogs Social Media*, 2013, pp. 1–4.

[15] M. Srivatsa and M. Hicks, "Deanonymizing mobility traces: Using social network as a side-channel," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*, 2012, pp. 628–637.

[16] 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 *Proc. Netw. Distrib. Syst. Secur. Symp.*, Feb. 2015, pp. 1–15.

[17] S. Nilizadeh, A. Kapadia, and Y.-Y. Ahn, "Community-enhanced de-anonymization of online social networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2014, pp. 537–548.

[18] S. Ji, W. Li, M. Srivatsa, J. S. He, and R. Beyah, "Structure based data de-anonymization of social networks and mobility traces," in *Proc. Int. Conf. Inf. Secur.* Cham, Switzerland: Springer, 2014, pp. 237–254.

[19] Y. Shao, J. Liu, S. Shi, Y. Zhang, and B. Cui, "Fast de-anonymization of social networks with structural information," *Data Sci. Eng.*, vol. 4, no. 1, pp. 76–92, Mar. 2019.

[20] J. Qian, X.-Y. Li, C. Zhang, and L. Chen, "De-anonymizing social networks and inferring private attributes using knowledge graphs," in *Proc. IEEE INFOCOM 35th Annu. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[21] S. Ji, T. Wang, J. Chen, W. Li, P. Mittal, and R. Beyah, "De-SAG: On the de-anonymization of structure-attribute graph data," *IEEE Trans. Depend. Sec. Comput.*, vol. 16, no. 4, pp. 594–607, Jul. 2019.

[22] H. Jiang, J. Yu, C. Hu, C. Zhang, and X. Cheng, "SA framework based de-anonymization of social networks," *Procedia Comput. Sci.*, vol. 129, pp. 358–363, Jan. 2018.

[23] S. Ji, W. Li, M. Srivatsa, and R. Beyah, "Structural data de-anonymization: Quantification, practice, and implications," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2014, pp. 1040–1053.

[24] J. Fang, A. Li, Q. Jiang, S. Li, and W. Han, "A structure-based de-anonymization attack on graph data using weighted neighbor match," in *Proc. IEEE 4th Int. Conf. Data Sci. Cyberspace (DSC)*, Jun. 2019, pp. 480–486.

[25] Z. Hu, L. Fu, and X. Gan, "De-anonymize social network under partial overlap," in *Proc. ACM Turing Celebration Conf. China*, May 2019, p. 16.

[26] J. Zhang, L. Fu, X. Wang, and S. Lu, "De-anonymization of social networks: The power of collectiveness," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Jul. 2020, pp. 89–98.

[27] X. Xian, T. Wu, S. Qiao, W. Wang, Y. Liu, and N. Han, "Multi-view low-rank coding-based network data de-anonymization," *IEEE Access*, vol. 8, pp. 94575–94593, 2020.

[28] M. Wang, Q. Tan, X. Wang, and J. Shi, "De-anonymizing social networks user via profile similarity," in *Proc. IEEE 3rd Int. Conf. Data Sci. Cyberspace (DSC)*, Jun. 2018, pp. 889–895.

[29] F. Shirani, S. Garg, and E. Erkip, "Optimal active social network de-anonymization using information thresholds," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 1445–1449.

[30] C. Zhang, H. Jiang, Y. Wang, Q. Hu, J. Yu, and X. Cheng, "User identity de-anonymization based on attributes," in *Proc. Int. Conf. Wireless Algorithms, Syst., Appl.* Cham, Switzerland: Springer, 2019, pp. 458–469.

[31] M. Naumov and T. Moon, "Parallel spectral graph partitioning," NVIDIA, Santa Clara, CA, USA, Tech. Rep., NVR-2016-001, 2016.

[32] J. Leskovec and A. Krevl. (2014). *SNAP Datasets: Stanford Large Network Dataset Collection*. [Online]. Available: http://snap.stanford.edu/data.2016:49

Honglu Jiang received the M.S. degree in computer science from Qufu Normal University, Shandong, China, in 2012, and the Ph.D. degree in computer science from The George Washington University, Washington, DC, USA, in 2021.

She is currently an Assistant Professor of computer science with The University of Texas Rio Grande Valley, Brownsville, TX, USA. Her research interests include wireless networks, differential privacy, big data, and privacy preservation.

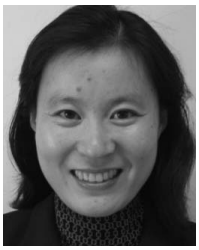
Jiguo Yu (Senior Member, IEEE) received the Ph.D. degree from the School of Mathematics, Shandong University, Qingdao, China, in 2004.

He became a Full Professor with the School of Computer Science, Qufu Normal University, Shandong, in 2007. He is currently a Full Professor with Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. His main research interests include privacy-aware computing, wireless networking, distributed algorithms, peer-to-peer computing, and graph theory including designing and analyzing algorithms for many computationally hard problems in networks.

Dr. Yu is a member of ACM and a Senior Member of the China Computer Federation (CCF).

Authorized licensed use limited to: George Mason University. Downloaded on January 31,2022 at 22:53:12 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



Xiuzhen Cheng (Fellow, IEEE) received the M.S. and Ph.D. degrees in computer science from the University of Minnesota—Twin Cities, Minneapolis, MN, USA, in 2000 and 2002, respectively.

She is currently a Professor of computer science with Shandong University, Qingdao, China. Her current research focuses on blockchain computing, privacy-aware computing, and wireless and mobile security.

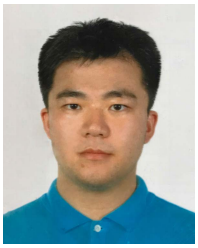
Dr. Cheng is a member of ACM. She served/is serving on the editorial boards of several technical journals and the technical program committees of various professional conferences/workshops. She was a Faculty Member with the Department of Computer Science, George Washington University, from September 2002 to August 2020, and worked as a Program Director for the US National Science Foundation (NSF) from April to October in 2006 (full time) and from April 2008 to May 2010 (part time). She received the NSF CAREER Award in 2004.

Cheng Zhang received the B.S. degree from Shandong Normal University, Jinan, Shandong, China, in 2015, and the M.S. and Ph.D. degrees in computer science from The George Washington University, Washington, DC, USA, in 2017 and 2020, respectively.

He is currently an Assistant Professor in computer information systems with West Texas A&M University, Canyon, TX, USA. His research interests include data anonymization and deanonymization, privacy-preserving in online social networks, and cooperative jamming in wireless networks.

Bei Gong received the B.S. degree from Shandong University, Qingdao, China, in 2005, and the Ph.D. degree from the Beijing University of Technology, Beijing, China, in 2012.

He has six National invention patents and one monograph textbook. He is the Principal Investigator of eight national projects such as the National Natural Science Foundation grants and six provincial and ministerial projects such as the General Science and Technology Program of Beijing Municipal Education Commission. Over the past five years, he has authored or coauthored more than 30 articles in top-tier journals and prestigious conferences in relevant research fields. His research interests include trusted computing, Internet of things security, mobile Internet of things, and mobile edge computing.



Haotian Yu received the B.A. degree from the University of Minnesota, Minneapolis, MN, USA, in 2017, and the M.S. degree in data analytics from The George Washington University, Washington, DC, USA, in 2020.

His research interest includes data analysis for social networks.

Authorized licensed use limited to: George Mason University. Downloaded on January 31,2022 at 22:53:12 UTC from IEEE Xplore. Restrictions apply.