Attribute-Enhanced De-anonymization of Online Social Networks

Cheng Zhang[1,2], Shang Wu[2], Honglu Jiang[1,2,3(✉)], Yawei Wang[2], Jiguo Yu[4], and Xiuzhen Cheng[1,2]

[1] School of Computer Science and Technology, Shandong University, Qingdao 266237, Shandong, People's Republic of China
[2] Department of Computer Science, The George Washington University, Washington, DC 20052, USA
{zhangchengcarl,swu23,hljiang0720,yawei,cheng}@gwu.edu
[3] School of Information Science and Engineering, Qufu Normal University, Rizhao 276826, Shandong, People's Republic of China
[4] School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, Shandong, People's Republic of China
jiguoyu@sina.com

Abstract. Online Social Networks (OSNs) have transformed the way that people socialize. However, when OSNs bring people convenience, privacy leakages become a growing worldwide problem. Although several anonymization approaches are proposed to protect information of user identities and social relationships, existing de-anonymization techniques have proved that users in the anonymized network can be re-identified by using an external reference social network collected from the same network or other networks with overlapping users. In this paper, we propose a novel social network de-anonymization mechanism to explore the impact of user attributes on the accuracy of de-anonymization. More specifically, we propose an approach to quantify diversities of user attribute values and select valuable attributes to generate the multipartite graph. Next, we partition this graph into communities, and then map users on the community level and the network level respectively. Finally, we employ a real-world dataset collected from Sina Weibo to evaluate our approach, which demonstrates that our mechanism can achieve a better de-anonymization accuracy compared with the most influential de-anonymization method.

Keywords: Online social network · Privacy · De-anonymization

1 Introduction

Nowadays, Online Social Networks (OSNs), such as Twitter and Instagram, are an integral part of daily life. According to the statistics revealed on [1], the world population reached 7.6 billion in January 2019. Over half the population used

A. Tagarelli and H. Tong (Eds.): CSoNet 2019, LNCS 11917, pp. 256–267, 2019.
https://doi.org/10.1007/978-3-030-34980-6_29

online social networks via websites and mobile applications every day. These active users provide a huge amount of valuable data including personal information and relationship among them for service providers. And such data also have a broad application area on academic research [10,14], business applications [18], homeland security [7], public health care [21], and so on [2]. Therefore, to protect the sensitive information of users while preserving the value of social network data, service providers usually publish "anonymized" social network data by removing the Personally Identifiable Information (PII, which are identifiable information to uniquely identify a user) while retaining user non-personally identifiable information (non-PII, or user attributes, e.g., gender, age, address), and modifying relationships before data publishing/sharing.

However, naive anonymized techniques cannot provide good protection, which have been proved to be vulnerable to de-anonymization attacks. Various de-anonymization attacks have been proposed to re-identify users in the anonymized social network by mapping them to the users in reference social network. Reference social networks include social relationships and real identities of users that can be collected by attackers via crawling the same social network or other social networks with overlapping users. Existing de-anonymization studies consider both the social network structure and attributes associated with users in social networks. However, most studies require a large number of "seeds" and often susceptible to a high noise ratio which represents the fraction of modified edges in an anonymized network.

Based on existing research, in this paper, we explore the impact of attribute values on users' privacy, and implement the multipartite graph consisting of users and attribute values in the anonymized network and the reference network to improve the accuracy of de-anonymization. Our contributions are summarized as follows:

- To the best of our knowledge, we are the first to perform de-anonymization attacks within a multipartite graph consisting of users and user attribute values in the anonymized network and the reference network.
- We propose an approach to quantify the attribute value diversity of each user attribute. This value is used to select valuable attributes from the anonymized network and the reference network to create the multipartite graph.
- Through extensive simulations on a real-world network dataset collected from the Sina Weibo, which is a famous social media in China, we suggest that our de-anonymization algorithm without the seeding phase is more robust to noise and can provide a significant improvement of accuracy compared to the baseline algorithm.

2 Related Work

Existing de-anonymization attacks could be divided into two main types, *structure-based de-anonymization attacks* and *attribute-attached de-anonymization attacks*.

Structure-based de-anonymization attacks aim to de-anonymize the anonymized social networks leveraging different structure (topology) similarities between the anonymized network and the reference network. This kind of attack has two branches, *seed-based de-anonymization attacks* and *seed-free de-anonymization attacks*.

- Seed-based de-anonymization attacks consist of *seed identification* phase and *propagation* phase. In the first phase, some users in the anonymized network are mapped to users with real identities in the reference network, and these mapped user pairs will serve as "seeds" in the next propagation phase. In the second phase, unmapped neighbors of seeds from the anonymized social network will be iteratively mapped to unmapped neighbors of seeds in the reference network using different structural similarity measurements, and the new mapped user pair will serve as a new seed pair for the next mapping iterations. In [14], Narayanan *et al.* proposed a de-anonymization algorithm based on social network topology to map users in an anonymized Twitter dataset to users in a Flickr dataset. Nilizadeh *et al.* [16] proposed a divide-and-conquer approach to de-anonymize the network from the community level to the entire network. Ji *et al.* [10] designed a De-Anonymization (DA) framework and an Adaptive De-Anonymization (ADA) framework based on proposed structural similarity, relative distance similarity and inheritance similarity. In [4], Chiasserini *et al.* proposed a degree-driven graph matching (DDM) algorithm with considering a social network to be represented by a Chung-Lu random graph [5].
- Seed-free de-anonymization attacks do not require pre-mapped user pairs as seeds to bootstrap the de-anonymization attacks. Pedarsani *et al.* [17] proposed a Bayesian model-based probabilistic framework to de-anonymize two networks. At first, users in each network are sorted by degree (number of neighbors) in descending order. Then, starting from mapping users with the highest degree by the bipartite matching, other users are iteratively mapped based on their degrees and distance to one user mapped in the previous round, until all users are mapped. Ji *et al.* [8,9] proposed an optimization-based de-anonymization (ODA) algorithm. ODA is a single-phase cold start algorithm and aims at minimizing the neighborhood's difference between an unmapped user in the anonymized network and an unmapped user in the reference network.

Attribute-Attached De-anonymization Attacks. By considering the impacts of user attributes (non-PII), which are published with the social network structure, various stronger attribute-attached de-anonymization attacks are proposed. Zhang *et al.* introduced a de-anonymization attack to heterogeneous information networks in [22]. They utilized attribute information in user entity matching and link matching to improve the accuracy of de-anonymization. In [12], Korayem and Crandall took a machine learning approach which employs various features based on temporal activity similarity, text similarity, geographic similarity, and social connection similarity to de-anonymize users across heterogeneous social computing platforms. Li *et al.* [13] took into account the structural

transformation similarity in social networks to propose an enhanced structure-based de-anonymization attack. In [19], Qian *et al.* presented that attacker's background information can be modeled by knowledge graphs to enhance the de-anonymization and attribute inference attacks. To de-anonymize Structure-Attribute Graph (SAG) data, Ji *et al.* [11] proposed a new de-anonymization framework called De-SAG which considered both the graph structure and the attribute information. In [23], Zhang *et al.* introduced an approach to quantify the significance of attributes in a social network. Then, based on the significance values of attributes, they proposed an attribute-based similarity measure to improve the social network de-anonymization performance.

3 Background

In this section, we introduce the definitions of the data model and three types of graphs as well as the attack model. Moreover, we introduce the community detection which is a crucial method for multipartite graph partition in our work. The mathematical notations used in this paper are summarized in Table 1.

3.1 Network Model

In this paper, we model the social network as an *undirected, unweighted, attributed and connected graph*. The terms "network", "user", and "link" are used interchangeably with "graph", "node", and "edge", respectively.

A graph, $G(V, E, A)$ consists of a set of users $V = \{v_1, v_2, ..., v_i, ...\}$ in social network, a set of edges $E = \{e_{i,j} = (v_i, v_j) | v_i, v_j \in V, i \neq j\}$ that represent social relationships between users, and a set of attributes (all the non-PII related to the users in V) $A = \{a_1, a_2, ..., a_i, ...\}$. Each attribute a_i has a set of attribute values denoted by $a_i = \{a_i^1, a_i^2, ..., a_i^j, ...\}$ (in order to simplify the discussion, all attribute values are discrete). $A(v_i)$ denotes the set of attribute values associated with user v_i. Given a graph G, it can be partitioned into a set of communities, which can be denoted by $C = (c_1, c_2, ..., c_i, ..)$. Furthermore, $|V|, |E|, |A|, |C|$ denote the number of users, edges, attributes and communities, respectively.

Given an original graph G, the anonymized G is denoted by $G_a = (V_a, E_a, A_a)$. In G_a, V_a is obtained by removing the PII from users ($V_a = V$, but the identities of users in V_a are indistinguishable). The edge set E_a is obtained by randomly adding and/or removing edges to/from E. The attributes (non-PII) associated with users are preserved in A_a, which means $A_a = A$ (it is also realizable to make $A \neq A_a$ by modifying attribute values from users in V during the anonymization process).

A reference graph denoted by $G_r = (V_r, E_r, A_r)$ can be obtained by crawling the same social network or different social networks with overlapping users, or by collecting from public databases.

A multipartite graph $G_m(V_m, E_m)$ is a graph whose nodes can be or are divided into several independent sets, $V_m = \{V_{m1}, V_{m2}, ..., V_{mn}\}$. E_m denotes the edge set.

Table 1. Notations

Symbol	Definition
G, G_a, G_r, G_m	Original, anonymized, reference, multipartite graphs
V, V_a, V_r	Node (user) set
E, E_a, E_r	Edge (relationship) set
A, A_a, A_r	Attribute set
$A(v_i), A_a(v_i), A_r(v_i)$	The set of attribute values associated with user v_i in G,G_a,G_r
v_i	The ith user
a_i	The ith attribute
a_i^j	The jth value of attribute a_i
C	Community set
c_i	The ith community
$\|V\|, \|E\|, \|A\|, \|C\|$	The number of users, edges, attributes and communities.
S_{a_i}	The set of users in V that possess the attribute a_i
Sa_i^j	The set of users possessing the attribute value a_i^j
$Deg(v_i)$	The degree of user v_i

3.2 Attack Model

Next, in the attack model, we assume that attackers can access two social graphs. One is the anonymized graph G_a including sensitive information associated with users in V_a. The other one is the reference graph G_r including the true identities associated with users in V_r. In these two graphs, we assume $V_r \cap V_a \neq \emptyset$, $E_a \cap E_r \neq \emptyset$, and $A_a = A_r$. Based on these assumptions, the attacker aims to map the users in G_a to those in G_r so that they can disclose the private information of users in G_a. This attack can be mathematically defined as a mapping from V_a to V_r [8,16]:

$$f : V_a \rightarrow V_r = \{(i, f(i) = j) | i \in V_a, j \in V_r\}. \quad (1)$$

3.3 Community Detection

Community structure (or clusters, groups) commonly exists in various types of networks, such as social networks, academic structures like research citations (Arxiv, Google Scholar), biological networks, etc. Members in the same community have a higher probability of being connected and are more likely to have interaction with each other than with other members from other communities. As one of the most popular research topics, community detection (or graph partitioning) has been a fundamental problem in exploring complex network structures and extracting valuable information.

Many community detection approaches have been proposed and widely used. In our work, we employ the modularity-based community detection algorithm [6]

to partition the multipartite graph into small groups, since it has a good balance between speed and accuracy, and has the ability to partition the multipartite graph into a set of small and dense communities without a predefined number of communities. In community detection, *modularity* [3,15] is a quality index to assess the quality of a network partition. Networks with high modularity have dense connections among the nodes within communities but sparse connections between nodes in different communities. Given an undirected connected graph $G(V, E)$, it can be partitioned into a set of communities C which can be used to calculate modularity $M(C)$ of G by the following equation:

$$M(C) = \sum_{c_i \in C} \left[\frac{|E_{c_i}|}{|E|} - \left(\frac{|E_{c_i}| + \sum_{c_j \in C} |E_{c_i,c_j}|}{2|E|} \right)^2 \right], \tag{2}$$

where $|E_{c_i}|$ denotes the number of edges in cluster c_i, and $|E_{c_i,c_j}|$ indicates the number of inter-community edges that connect one node in community c_i with the other node in community c_j, and $c_i \neq c_j$.

4 Attribute-Enhanced De-anonymization Scheme

The motivation of our work is that attributes with diverse values could better represent a user if those values of attributes are widely distributed in a social network.

Network communities can offer an efficient way to divide and conquer the de-anonymization attack. Based on the attribute with diverse values, the anonymized graph and the reference graph could be merged into one multipartite graph which provides a fresh idea to conduct de-anonymization attacks. After dividing the multipartite graph into small communities, some users inside each community are mapped first, and then the remaining users in the anonymized graph are mapped to users in reference graph based on the global propagation which runs on the whole network.

Figure 1 illustrates our approach which has four steps: (1) multipartite graph generation, (2) multipartite graph partitioning, (3) local mapping, and (4) global propagation.

4.1 Attribute Value Diversity (AVD)

We first define $AVD(a_i)$ which indicates the diversity of attribute value for each attribute in a social network. This definition borrows the concept of information entropy. Thus, $AVD(a_i)$ is measured as follows:

$$AVD(a_i) = -\sum_{j=1}^{|a_i|} \left(\frac{|Sa_i^j|}{|S_{a_i}|} \times \ln \frac{|Sa_i^j|}{|S_{a_i}|} \right), \tag{3}$$

where a_i is the ith attribute in A, $|S_{a_i}|$ denotes the number of users in V that possess the attribute a_i in A, and $|Sa_i^j|$ represents the number of users possessing the value of a_i^j in a_i. $|a_i|$ denotes the number of different attribute values

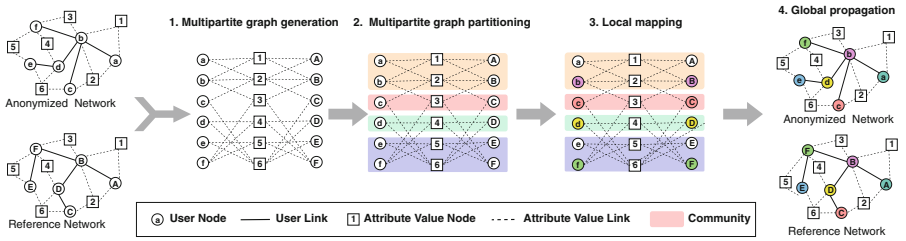


Fig. 1. An overview of our approach: (1) the anonymized network and reference network are merged into one multipartite graph; (2) the multipartite graph are divided to small communities; (3) users inside each community are mapped; and (4) the remaining users are mapped by performing a global propagation on the whole network.

a_i^j of attribute a_i. A large $AVD(a_i)$ indicates that attribute a_i has large diverse attribute values. For example, a user in a social network has two attributes: *Gender* and *Address*. For each user, gender only has two possible attribute values (male and female), while address could be different from one to the other. A fine-grained address, such as the detailed mailing address, could more precisely identify a user than the gender. Therefore, we claim that $AVD(Address)$ is larger than $AVD(Gender)$. In addition, as an important step in our de-anonymization approach. We apply the attribute value diversity to generate a multipartite graph.

4.2 Node Similarity

In mapping step and global propagation step, the node v_a in V_a is mapped to node v_r in V_r based on the node similarity $sim(v_a, v_r)$, which is defined as follows:

$$sim(v_a, v_r) = \frac{|A(v_a) \cap A(v_r)|}{|A(v_a) \cup A(v_r)|} \times (1 - \frac{|Deg(v_a) - Deg(v_r)|}{\sqrt{Deg(v_a) * Deg(v_r)}}), \tag{4}$$

where $\frac{|A(v_a) \cap A(v_r)|}{|A(v_a) \cup A(v_r)|}$ measures the similarity of attribute values between node v_a and node v_r; $Deg(v_a)$ and $Deg(v_r)$ denote the degree of v_a and v_r in G_a and G_r, respectively. As a result, the similarity between two nodes is determined by their attribute information and structural characteristics.

4.3 Algorithm Details

Multipartite Graph Generation: The first step of our approach is to create a multipartite graph G_m. As shown in Algorithm 1, given an anonymized graph $G_a(V_a, E_a, A_a)$ and a reference graph $G_r(V_r, E_r, A_r)$, their user sets V_a, V_r are added to G_m as nodes sets V_{m1} and V_{m2}, respectively (Line 1–3). For any attribute a_i, it will be removed from A_a and A_r if its diversity of attribute value $AVD(a_i)$ is smaller than a threshold (Line 4–13), because adding attributes

with few attribute values into the multipartite graph would make the graph too complex to get a good graph partitioning quality. This threshold depends on the number of attributes owned by graphs. For instance, a graph with abundant attributes would select a large threshold. After that, attribute values of each user in V_a and V_r serve as a set of nodes V_{m3} and are added into G_m, respectively. Meanwhile, if a user possesses an attribute value, an edge should be added between the user and the attribute value (Line 14–25). Finally, the multipartite graph consisting of user nodes and attribute value nodes are created.

Multipartite Graph Partitioning: As mentioned in Sect. 3.3, we employ modularity-based community detection algorithm [6]. The partitioning operation will keep repeating until no community can be split further (or reaching a maximum number of iterations).

Algorithm 1: Creating a multipartite graph

Input : Two social graphs: $G_a(V_a, E_a, A_a)$ and $G_r(V_r, E_r, A_r)$
Output: A multipartite graph $G_m(V_{m1}, V_{m2}, V_{m3}, E_m)$

1 Set an undirected graph $G_m = \emptyset$
2 $G_m.add\_nodes\_from(V_a, type = V_{m1})$
3 $G_m.add\_nodes\_from(V_r, type = V_{m2})$
4 **for** *each attribute* $a_i \in A_a$ **do**
5 **if** $AVD(a_i) < threshold$ **then**
6 $A_a.remove(a_i)$
7 **end**
8 **end**
9 **for** *each attribute* $a_i \in A_r$ **do**
10 **if** $AVD(a_i) < threshold$ **then**
11 $A_r.remove(a_i)$
12 **end**
13 **end**
14 **for** *each user* $v_i \in V_a$ **do**
15 **for** *each attribute value* $a_i^j \in A_a(v_i)$ **do**
16 $G_m.add\_node(a_i^j, type = V_{m3})$
17 $G_m.add\_edge((v_i, a_i^j), type = E_m)$
18 **end**
19 **end**
20 **for** *each user* $v_i \in V_r$ **do**
21 **for** *each attribute value* $a_i^j \in A_r(v_i)$ **do**
22 $G_m.add\_node(a_i^j, type = V_{m3})$
23 $G_m.add\_edge((v_i, a_i^j), type = E_m)$
24 **end**
25 **end**

Local Mapping: After graph partitioning, each community $c_i \in C$ has three sets of nodes which are V'_{m1},V'_{m2}, and V'_{m3}. These three sets are the subset of V_{m1}, V_{m2} and V_{m3}, where V_{m1} contains V_a, V_{m2} contains V_r and V_{m3} attribute values, respectively. In each community, an unmapped user v_i in V'_{m1} is mapped to a candidate user v_j in V'_{m2}, based on the node similarity calculated by Eq. (4) in Sect. 4.2. Furthermore, during the local mapping, we employ the following eccentricity [14] to measure the uniqueness of the unmapped user candidate:

$$ecc(D) = \frac{\max_1(D) - \max_2(D)}{\delta(D)}, \tag{5}$$

where D is the list of similarity scores sim between v_i and all candidate users in V_{m2}, $\max_1(D)$ and $\max_2(D)$ are the two highest similarity values in D, and $\delta(D)$ represents the standard deviation of the values in D. If the $ecc(D)$ exceeds a threshold, the users v_i is mapped to v_j with the similarity equals to $\max_1(D)$.

Algorithm 2: Local mapping

```
Input  : A set of communities: C
Output: A set of mapped user pairs M
Initialize M = ∅
for each community c_i ∈ C do
    c_i = V'_m1 ∪ V'_m2 ∪ V'_m3, where V'_m1 ⊂ V_m1, V'_m2 ⊂ V_m2, V'_m3 ⊂ V_m3
    for each v_i ∈ V'_m1 do
        Initialize D = ∅
        for each v_j ∈ V'_m2 do
            Calculate sim(v_i, v_j) based on Equation (4)
            D.add(sim(v_i, v_j))
        end
        Calculate ecc(D) based on Equation (5), if it is above a threshold,
        select the user v_j with the highest sim(v_i, v_j)
        M = M ∪ {(v_i, v_j)}
    end
end
```

Global Propagation: This phase is similar to the propagation proposed in [14]. Staring from the identified users M from local mapping, each unmapped user v_a in G_a will be mapped to an unmapped user v_r in G_r. At each iteration, we randomly pick an unmapped user v_a who has a successfully mapped neighbor from anonymized social graph G_a, and use similarity measurement Eq. (4) to quantify its similarity values with all unmapped users candidates in G_r who possess at least one successfully mapped neighbor. Finally, the v_r will be selected based on the eccentricity defined in Eq. (5) to map with v_a. The new mapped pair (v_a, v_r) will be added into M to serve the next round iteration.

5 Experiments

In this section, we employ a real-world dataset collected from Sina Weibo, the most famous social media in China, to evaluate our attribute-enhanced de-anonymization approach.

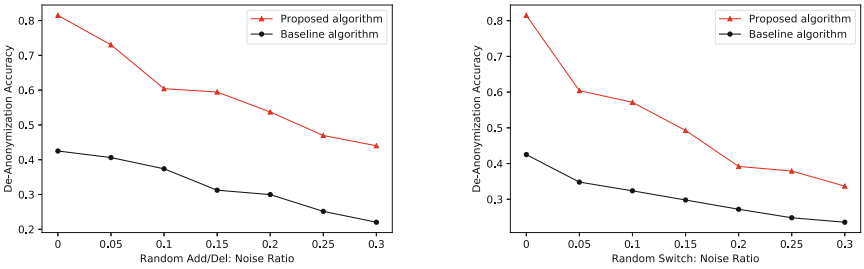
5.1 Experimental Setup

We converted Weibo dataset which captures the "following" relationships among users into an undirected graph with 3859 nodes, 4992 edges and an average degree of 2.587. Each user in this dataset possesses three attributes including "gender", "city" and "province". We make a copy of the original graph and replace all user identities with random characters, then employ two edge randomization methods [20], Random Add/Del and Random Switch on this copy to generate anonymized graphs. Next, we duplicate the original graph and randomly remove 10% of nodes and their edges to generate the reference network. In our work, we used the most influential de-anonymization approach [14] as the baseline. The de-anonymization accuracy is used to evaluate the de-anonymization performance, which denotes the ratio of the number of users correctly re-identified over the number of overlapping users of G_a and G_r. The following evaluation results are the average of 30 trials.

5.2 Results

At first, we apply two edge randomization approaches to add noises into the anonymized network. The first approach is Random Add/Del, which randomly deletes a number of edges from a network and then randomly adds the same number of edges into the network. The second approach called Random Switch randomly removes two edges $e_{i,j}$ and $e_{p,q}$ from network and then add two edges $e_{i,p}$ and $e_{j,q}$ to the network. In evaluations, the *noise ratio*, which decides the number of added and deleted edges over the total number of edges in the network, is changed from 0 to 0.3 at an interval of 0.05. Based on experiences from our experiments, the threshold used in multipartite graph generation is set to the average value of AVD, and the maximum number of partitioning iterations is set to 6. The eccentricity threshold used in local mapping is set to 0.1 [16], and the number of seeds in the baseline algorithm is set to 150 [14].

Figure 2a depicts the result of de-anonymization accuracy in the anonymized network processed by Random Add/Del. As a result of using degrees of nodes to calculate a similarity between users, both de-anonymization methods are negatively affected by adding and removing edges, compared to the case without adding noise. Despite this, our algorithm has better performance than the baseline. Figure 2b presents the result in the network processed by Random Switch. As shown in the result, we can see that both our algorithm and the baseline are impacted by added noises. This is because even though the degrees of nodes are not changed by Random Switch, the inside structure of the network which both de-anonymization algorithms rely on is disturbed. In spite of this, our algorithm is still better than the baseline.



(a) Random Add/Del anonymized graph. (b) Random Switch anonymized graph.

Fig. 2. Impact of noise on the de-anonymization accuracy in the different anonymized graphs

6 Conclusion and Future Work

In this paper, we propose a new approach to de-anonymize a social network with user attributes. Our method merges the anonymized network and reference network into a multipartite graph and employs a modularity-based community detection technique to partition the multipartite graph into small and dense communities. After that, the local mapping and global propagation are performed sequentially to de-anonymize users from the community level and the network level, respectively. We use a Sina Weibo dataset processed by two edge randomization methods to evaluate our algorithm. The evaluation results indicate that our approach is more efficient than the most influential algorithm. In the future, we would explore ways to de-anonymize the networks with modified attribute values and measure the degree of anonymity of users in an anonymized network with user attributes.

Acknowledgment. This work was partially supported by the US National Science Foundation under grant CNS-1704397, and the National Science Foundation of China under grants 61832012, 61771289, and 61672321.

References

1. The global state of digital in 2019 report, January 2019. https://hootsuite.com/pages/digital-in-2019
2. The U.S. governments open data, July 2019. https://www.data.gov
3. Brandes, U., et al.: On modularity clustering. IEEE Trans. Knowl. Data Eng. **20**(2), 172–188 (2007)
4. Chiasserini, C.F., Garetto, M., Leonardi, E.: Social network de-anonymization under scale-free user relations. IEEE/ACM Trans. Netw. **24**(6), 3756–3769 (2016)
5. Chung, F., Lu, L.: The average distance in a random graph with given expected degrees. Internet Math. **1**(1), 91–113 (2003). https://projecteuclid.org:443/euclid.im/1057768561
6. Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70**(6), 066111 (2004)

7. Hayes, B.: Connecting the dots. Am. Sci. **94**(5), 400–404 (2006)
8. Ji, S., Li, W., Srivatsa, M., Beyah, R.: Structural data de-anonymization: quantification, practice, and implications. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 1040–1053. ACM (2014)
9. Ji, S., Li, W., Srivatsa, M., Beyah, R.: Structural data de-anonymization: theory and practice. IEEE/ACM Trans. Netw. **24**(6), 3523–3536 (2016)
10. Ji, S., Li, W., Srivatsa, M., He, J.S., Beyah, R.: General graph data de-anonymization: from mobility traces to social networks. ACM Trans. Inf. Syst. Secur. (TISSEC) **18**(4), 12 (2016)
11. Ji, S., Wang, T., Chen, J., Li, W., Mittal, P., Beyah, R.: De-SAG: on the de-anonymization of structure-attribute graph data. IEEE Trans. Dependable Secure Comput. (2017)
12. Korayem, M., Crandall, D.: De-anonymizing users across heterogeneous social computing platforms. In: Seventh International AAAI Conference on Weblogs and Social Media (2013)
13. Li, H., Zhang, C., He, Y., Cheng, X., Liu, Y., Sun, L.: An enhanced structure-based de-anonymization of online social networks. In: Yang, Q., Yu, W., Challal, Y. (eds.) WASA 2016. LNCS, vol. 9798, pp. 331–342. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42836-9_30
14. Narayanan, A., Shmatikov, V.: De-anonymizing social networks. arXiv preprint arXiv:0903.3276 (2009)
15. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (2004)
16. Nilizadeh, S., Kapadia, A., Ahn, Y.Y.: Community-enhanced de-anonymization of online social networks. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 537–548. ACM (2014)
17. Pedarsani, P., Figueiredo, D.R., Grossglauser, M.: A Bayesian method for matching two similar graphs without seeds. In: 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 1598–1607. IEEE (2013)
18. Perez, S.: Twitter partners with IBM to bring social data to the enterprise. Tech Crunch (2014)
19. Qian, J., Li, X.Y., Zhang, C., Chen, L.: De-anonymizing social networks and inferring private attributes using knowledge graphs. In: The 35th Annual IEEE International Conference on Computer Communications, INFOCOM 2016, pp. 1–9. IEEE (2016)
20. Ying, X., Wu, X.: Randomizing social networks: a spectrum preserving approach. In: Proceedings of the 2008 SIAM International Conference on Data Mining, pp. 739–750. SIAM (2008)
21. Young, S.D.: A big data approach to HIV epidemiology and prevention. Prev. Med. **70**, 17–18 (2015)
22. Zhang, A., Xie, X., Chang, K.C.C., Gunter, C.A., Han, J., Wang, X.: Privacy risk in anonymized heterogeneous information networks. In: EDBT, pp. 595–606. Citeseer (2014)
23. Zhang, C., Jiang, H., Wang, Y., Hu, Q., Yu, J., Cheng, X.: User identity de-anonymization based on attributes. In: Biagioni, E.S., Zheng, Y., Cheng, S. (eds.) WASA 2019. LNCS, vol. 11604, pp. 458–469. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-23597-0_37