Anonymization in Online Social Networks Based on Enhanced Equi-Cardinal Clustering

Madhuri Siddula, Yingshu Li, Xiuzhen Cheng, *Fellow, IEEE*, Zhi Tian, and Zhipeng Cai

***Abstract*— Recent trends show that the popularity of online social networks (OSNs) has been increasing rapidly. From daily communication sites to online communities, an average person's daily life has become dependent on these online networks. Hence, it has become evident that protection should be provided to these networks from unwanted intruders. In this paper, we consider the data privacy on OSNs at the network level rather than the user level. This network-level privacy helps us to prevent information leakage to third-party users, such as advertisers. We propose a novel scheme that combines the privacy of all the elements of a social network: node, edge, and attribute privacy by clustering the users based on their attribute similarity. We use an enhanced equi-cardinal clustering (ECC) as a way to achieve k-anonymity. We further improve k-anonymity with l-diversity. Our proposed enhanced ECC ensures that there are at least "k" users in any given network as well as the attributes in each cluster has at least l-distinct values. We further provide proofs on how the proposed ECC ensures k-anonymity and the maximum information loss. We consider a weighted directed social network graph as an input to our method to consider the existing complexities in a social network. With the help of two real-world data sets, we evaluate this method in terms of privacy and efficiency.**

***Index Terms*— Anonymization, clustering, equicardinal, online social network (OSN), privacy.**

I. Introduction

SOCIAL media has become a way of communication in today's world. Social network penetration worldwide is ever-increasing. According to [1], in 2017, 71% of internet users were social network users, which is 2.62 billion users. With over 1.86 billion monthly active users, Facebook is currently the market leader in terms of reach and scope. It has become the way of communication both directly and indirectly. While direct communication happens through chatting, indirect communication happens through posts on the network. These posts are generally made public for everyone to see and are the attracting points to various attackers. People share the places they have visited, restaurants, movies, travel information, as well as personal photos and thoughts. Although this information is private to one's friend list, many social networking sites allow this information to be accessed by the third-party advertisers for profits. Advertisers use this information to broadcast tailored advertisements. Although this process seems innocuous, it might be misused when the information leak happens to the unwanted attackers [2]–[4]. Some of the most common attacks are behavioral advertising, identity theft, and online stalking.

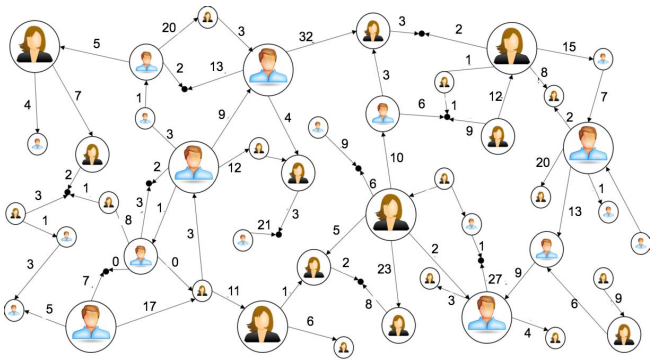

Fig. 1. Social network as a graph.

As shown in Fig. 1, any social network can be represented as a graph, where each user profile can be represented as a node and friendship between the two users is represented by an edge between those two nodes. As can be seen from Fig. 1, a node can have multiple edges: user to user nodes, a user to attribute nodes, and attribute to attribute nodes. Hence, we have three ways to anonymize the social network graph, i.e., we can anonymize users, links, and attributes. However, the main aim of any anonymization technique is that it should not remove too much information that causes structural information loss (IL) and utility of the original graph.

One of the significant concerns in protecting the privacy of online social networks (OSNs) is identity disclosure. A simple, naive anonymization technique is to replace a user's name with random identifiers. However, the intruders can exploit the structural information of such an anonymized graph. For example, in a given neighborhood, such as a school department, the node(s) with the highest number of edges can be identified as the head of the department with confirmation from sensitive attributes, such as age, health condition, and salary. Hence, the network should be anonymized in such a way that trusting advertisers can benefit from it but still is useless to anyone who wants to steal personal information of individual users.

There are two crucial details in a network that should be protected: 1) information about a user's sensitive attributes and 2) connection/edge information of the users. While sensitive

Manuscript received January 1, 2019; revised May 11, 2019; accepted July 2, 2019. Date of publication July 25, 2019; date of current version August 8, 2019. This work was supported in part by the NSF under Grant 1704287, Grant 1252292, Grant 1741277, Grant 1829674, Grant 1704274, Grant 1704397, and Grant 1912753. *(Corresponding author: Zhipeng Cai.)*

M. Siddula, Y. Li, and Z. Cai are with the Department of Computer Science, Georgia State University Atlanta, GA 30302 USA (e-mail: zcai@gsu.edu).

X. Cheng is with the Department of Computer Science, George Washington University, Washington, DC 20052 USA.

Z. Tian is with the Department of Computer Engineering, George Mason University, Fairfax, VA 22030 USA.

Digital Object Identifier 10.1109/TCSS.2019.2928324

attributes can be removed during anonymizing, edge information is hard to eliminate. Most of the advertisers depend on the user-attribute edge information to send personalized advertisements. Hence, we need an anonymization technique that protects the details, as mentioned earlier in a given network.

Researchers have studied the mechanisms of protecting nodes from identity disclosure when an adversary has background information about the graph structure around a node of interest. In these approaches, each node has structural properties that are the same as the ones of a small set of other nodes in the graph [5], [6]. Often achieving neighborhood similarity requires considerable structure distortion. Link disclosure refers to inference about the existence of a link between two known individuals and in the focus of this paper. Since complete removal of the links to keep structural properties private would yield a disconnected graph, the edge modification technique proposes edge addition and deletion to meet the desired constraint [7]. k-isomorphism [8], [9] is a way to achieve such structural similarity, where the original graph is transformed into "k" disconnected pairwise isomorphic subgraphs through link insertion and deletion. Another approach proposed an edge modification algorithm that achieves k-automorphism anonymity [10]. In general, considerable structural distortion is required to provide the desired structural similarity.

In this paper, we propose an enhanced equi-cardinal clustering (ECC) method to anonymize the users of a given network. A simple clustering method ensures that there is no more individual information that can be retrieved from the anonymized network. This mechanism ensures both vertex and edge anonymization without the loss of much information. All similar users can be grouped into a single cluster. For example, all the Ph.D. students from the computer science department in a given city can be grouped. This ensures that the common (non-sensitive) attributes of all such users are now represented with a single cluster head. Assume a scenario where a user's profile is nothing, but their location information that is their current address, for example, ⟨Bob, Apt 1234, 123 Main St., Bay Area, California, USA⟩. If the user uses this information for any location-based service (LBS) query, his address can be retrieved by an attacker. However, by using the proposed method, we find "k" users of the social network whose geographical locations are close. Assume all of them live in the Bay Area but might be on different streets and different apartments. The generalized profile now would be ⟨1, Bay Area, California, USA⟩. Using this generalized profile, the user asks for an LBS query. Thus, even if the attacker retrieves this information, there are "k" other users living in the same area, and also the exact address is still protected for all the users. However, the results might vary, that is, the utility drops. Hence, we aim to preserve the utility and provide the highest obfuscation. However, it is to be noted that a simple clustering method will not ensure that all the clusters are of equal size. This is to ensure that there are no information leaks due to the difference in the anonymization between the users. Hence, we enhance a simple clustering algorithm, such as k-means, to ensure that every cluster contains a minimum of "k" users. This, in turn, also ensures that each user cannot be distinguished from "$k-1$" similar users and, hence, guarantees k-anonymity.

As a method of clustering data, the k-means algorithm is widely used because of its simplicity and ability to converge extremely quickly in practice. Hay *et al.* [5] applied structural generalization approaches that groups nodes into clusters, by which privacy details about individuals can be hidden properly. Reasons for choosing k-means clustering (KMC) is explained in detail in Section IV. To ensure node anonymity, they proposed to use the size of a partition as a basic guarantee against re-identification attacks. However, the solution introduces considerable uncertainty in the released networks.

Following are our contributions.

1) To the best of our knowledge, we are the first to consider the privacy of all the elements of a network (nodes, edges, and attributes) while proposing an anonymization technique.
2) This paper ensures that all k-anonymities are maintained for all the users in the network.
3) Proposed enhancement for l-diversity ensures that there are no sensitive attributes in the generalized attribute set of the cluster head.
4) This paper also takes into consideration the knowledge graph attack and provides defense against it.

The rest of this paper is organized as follows. Section II reviews the related work on the privacy-preserving OSN models. The problem statement is defined in Section III. Proposed novel ECC is explained in Section IV. The performance metrics and the privacy-preserving analysis of our algorithm are provided in Section V. Section VI discusses the computational analysis. In Section VII, we evaluate the algorithm experimentally using the real-world data sets. Finally, conclusions and future work are discussed in Section VIII.

II. Related Work

Privacy in OSNs is a new research area that is still under development. Most of the research in this field is based on a computing perspective. We are discussing some of the methods that were proposed in the area and that are relevant. The predominance of the OSN privacy research considers an OSN as a network with nodes and links. According to [11]–[13], there are three areas, in which privacy has to be maintained to provide OSN privacy: node, link, and attribute privacy.

Node anonymization is achieved in different ways in the past. A naive and straightforward anonymization technique is to replace all the nodes with random numbers and alphabets [6], [14], [15]. Perturbation is another technique, in which we add or delete existing nodes to achieve the required anonymity [6]. As this simple method is prone to infiltration, there are more sophisticated algorithms proposed. If two nodes are structurally similar, then they are said to be automorphically equivalent. Automorphic equivalence partitions the graph, whose members have identical properties. This ensures that even if an adversary gets hold of these subgraphs, they cannot be distinguishable among the users in that subgraph. Thus, the anonymity is obtained by the "hidden in the crowd" technique.

Link anonymization can also be achieved in various ways. A similar perturbation technique can be done to achieve link privacy [16]–[18]. Another way to achieve link anonymization is through random walks [19], [20]. In this technique, a random initial node and direction are selected. With the initial node and direction, a walk is performed until the desired length is reached. At the end of this walk, a new edge is added. Dwork [16], Dwork *et al.* [17], and Krishnamurthy and Wills [21] have introduced the Laplacian noise to achieve link anonymization. Methods, such as in [22]–[24], use the neighborhood as a measure to obtain links in the neighborhood and perturb them.

Attribute anonymization, though have not been explored heavily, is an important part in achieving social network privacy. Users have sensitive attributes that need to be protected even if the node and links have been anonymized. These attributes cannot simply be removed, as they provide important information for advertisers. All the anonymization techniques proposed previously have been summarized in [25]–[27].

Structural similarity is one of the best methods in achieving network anonymity. To achieve structural similarity, we need to ensure that the proposed method should follow k-anonymity [28]–[30]. k-anonymity ensures that the given user is indistinguishable among "k" other users. If the chosen "k" is large enough, we maintain the privacy of the user. A graph is called k-anonymous if there exist at least "$k-1$" users in the graph with the same degree as a given user. Zhou and Pei [31] proposed one of the earliest methods involving k-anonymity. This method selects nodes based on a cost function and, randomly, adds and removes edges to achieve the desired degree. Other methods proposed in [6] and [32] have also concentrated on adding and removing edges in different ways. Some of the method that utilized the concept of k-anonymity include [33]–[37]. We chose this metric in our method as it is a simple and effective way of achieving user privacy. l-diversity is another mechanism to preserve the privacy that is been proposed in [38]. Domingo-Ferrer and Torra [39] have focused on anonymizing a database using l-diversity principle, in which it overcame the problems of k-anonymity.

Knowledge graph [40] is the most effective way of de-anonymizing a graph that has been anonymized using the above-described techniques. This technique considers that an attacker can obtain knowledge from many sources, including common sense, publicly available data, personal information, and network structural data. When all such information is combined, we form a knowledge graph that can be used to infer some of the data from the anonymized graph.

There are very few methods that have considered clustering to achieve k-anonymity preserving privacy. Skarkala *et al.* [24], Liu and Yang [41], Liu and Terzi [42], and Kayes and Iamnitchi [43] have done node grouping that takes into consideration of edge weights. Although edge weights are an important metric for clustering, this method(s) does not ensure that the users with similar qualities are grouped. For advertisers, two users with a similar profile can be seen as a single node rather than users with the same number of friends. In other methods, such as in [44] and [45], authors have applied perturbation along with aggregation. In this method, simple clustering methods are combined with perturbation to maintain the graph structural similarity. This method does not ensure the equal division of users among clusters as well as remove the critical information. Hay *et al.* [5] have proposed partitioning methods based on the topology of a given graph. This method does not consider user profile for clustering either.

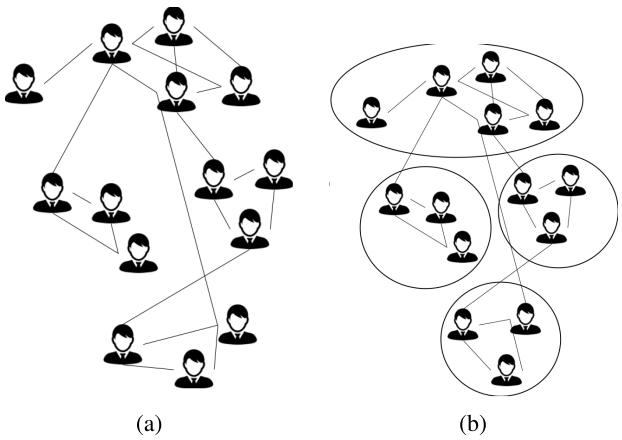

Fig. 2. Users connected in a network are clustered based on attribute distance. (a) Users connected in a network. (b) Clustered users.

Zhou and Pei [31] and Zhang *et al.* [46] have enhanced the traditional k-anonymity and l-diversity methods to anonymize the social networks using both node and link perturbations. All these works have either considered node or edge privacy but not both. Also, attribute anonymization has not explicitly been proposed.

III. Problem Statement

Let us assume that we have a social network graph $G = (V, E)$, where V represents the users and E represents the edges between them. $V = \{u_1, u_2, u_3, \ldots, u_n\}$ and $|V| = n$. Every user, u_i, has a set of attributes, represented by A, values associated with it $\{a_{i1}, a_{i2}, \ldots, a_{im}\}$. Edges are directed and, hence, are represented as $E = \{e_{12}, e_{21}, e_{13}, \ldots, e_{n,n-1}\}$ and $|E| = n * (n-1)$. Here, an edge e_{12} represents an edge from vertex 1 to vertex 2, which is not same as e_{21}, an edge from vertex 2 to vertex 1. In addition, every edge has a corresponding weight.

This paper aims to anonymize graph G, such that the anonymized graph $G = (V, E)$ with anonymized vertex set V', anonymized edge set E', and anonymized attribute set A' follows two objectives.

1) Given a constant k, for each node "u," the probability that an attacker re-identifies u is at most $(1/k)$.
2) Minimize IL while maximizing the degree of anonymity.

This paper follows the clustering method to cluster users, which are closely related, to form meaningful clusters of users. The users belonging to a single cluster, let us say "k" users, will be represented with a single cluster head "c." This clustering allows us to anonymize "k" users with a single node, thus allowing the k-anonymity. This can be seen in Fig. 2.

IV. Proposed Method

The proposed method aims to cluster users whose profiles are similar. For example, two students studying at University A

should be grouped than two students studying in different universities. KMC is one such method that groups users based on their closeness. The main advantage of k-means is that it is fast and produces tighter clusters. The k-medoids algorithm, a similar yet powerful technique, is an alternative to a k-means algorithm, where we need to reduce the sum of squared distances (SSD) between the cluster points and their corresponding cluster heads. Although k-medoids give us a smaller SSD value than k-means, it will not provide us better privacy in our scenario. This is because k-medoids are calculated based on the principle that a medoid is calculated for a cluster rather than mean. A medoid is one of the data points in a cluster. In our scenario, data points represent the user's profile. Thus, selecting a medoid as the cluster head indicates that we are choosing one of the user's profiles as the cluster head. Thus, we are exposing the exact user's attribute values. However, the main aim of this paper is precisely the opposite. In addition, since the k-values are not large, a simple brute force attack will reveal the exact user to which this profile is linked.

Another advantage with k-means is its tolerance to outliers. We have used a k-means algorithm to consider outliers as a part of our system. In the problem statement, we have mentioned that the network we use contains user's profiles that have been scanned for any possible Sybil users. As there will not be any Sybil users in the system, all the users are genuine users of the social network. Hence, we need to provide the same privacy level to everyone. If we use a clustering technique that leaves out outliers, it means we are leaving out users just because they live in a far area, or their likes and dislikes are different from other users. However, in fact, such users require more privacy than the rest and the attacker who identifies that a user lives in a far area can be vulnerable to theft. Hence, using k-means avoid ignoring outliers and provides similar privacy for all of them.

Once the initial clusters are obtained using k-means, we perform our ECC method to produce clusters of equal size. To cluster users, we need a distance metric to calculate how near or far users are. Section IV-A describes the method to compute the distance between any two given users.

Knowledge graph attacks can be avoided with the proposed method, as the attributes are generalized. Therefore, even if the attacker has a background knowledge, no individual in the anonymized graph has a single attribute whose values are the same as their exact attribute value.

A. Computing Distance Matrix

In this method, we explain how a distance matrix is calculated. Our initial step is to compute the clusters based on the KMC algorithm. To calculate distance as a part of k-means, we use the attributes between users. Once the algorithm finishes on the initial data, we have clusters with the users. We use the user attribute mean as the attributes for these initially formed clusters. Then, we have to calculate the distance of each user again with the cluster centroids as a part of re-assigning them. Following method describes how to calculate such a distance matrix.

Let us assume that the user vector $|\overrightarrow{u_i}|$ represents all the attributes associated with user u_i, and let "R" is the number of attributes. Similarly, let $|\overrightarrow{u_i}|$ and $|\overrightarrow{c_j}|$ be user "i" and centroid "j." Distance between them can be calculated in an R-dimensional space as

$$D_{i,j}^2 = |P_i - C_j|^2 = \sum_{n=1}^{R} (p_{i,n} - c_{j,n})^2. \tag{1}$$

One problem by clustering the users using KMC is that the resulting clusters can be of uneven size. This is a significant problem in our scenario, as not all users are protected with the same level of privacy. Hence, we need clusters of equal size to ensure the even privacy distribution over all users. Hence, to achieve overall k-anonymity, let us construct a distance matrix in ascending order. Let D_A represents this ascending-ordered distance array. The values in this array are described in the following:

$$\begin{pmatrix} D_{i_1,j_1} \\ D_{i_2,j_2} \\ D_{i_3,j_3} \\ . \\ . \\ . \\ D_{i_m,j_m} \end{pmatrix}. \tag{2}$$

The array D_A is of size $nk * 1$. This method is explained in detail in Algorithm 1.

Algorithm 1 Calculate Distance Matrix

```
Input: N → Set of users, C → Set of cluster centroids, A
    → Set of attributes
Output: D_A → Ordered distance matrix
for do i = 1 to n
    for do j = 1 to k
        D(u_i, c_j) = Σ_{a∈A} dist(u_ia, c_ja) / |A|
    endfor
endfor
m ← 1;
for i do = 1 to n
    for j do = 1 to k
        D_A(m, 1) ← D(u_i, c_j)
        D_A(m, 2) ← i
        D_A(m, 3) ← j
        m ← m + 1
    endfor
endfor
D_A ← MergeSort(D_A) based on first column
```

B. Equi-Cardinal Clustering

In the proposed method, the main aim is to anonymize all the users in the network in such a way that they cannot be distinguished among "k" other users. To achieve this, we should make sure that the anonymity of each user is greater than or equal to "k." In our method, we are achieving this anonymization using clustering. Also, "k" is calculated based on the best utility, that is, the value of "k" is determined based

on the number of users, their attributes, and the values of the attributes. The anonymization degree of the user u_i is the number of users in the cluster that u_i belongs to. To maintain k-anonymity, we should make sure that each cluster has at least "k" users. However, let us assume that there are two clusters C_1 and C_2. C_1 has "k" users, and C_2 has "$l = 10 * k$" users. Then, all the users in C_1 are k-anonymous, while users in C_2 are l-anonymous. This difference in anonymization level increases the probability of information leakage. Hence, to prevent any such information leaks, we maintain equal anonymity levels to all the users. Therefore, we create clusters that are of almost the same size as that of ECC.

To achieve ECC, we rearrange the users into different clusters. The primary aim of this method is to rearrange users into similar size clusters with minimal increase in IL. Hence, we aim at removing users from the clusters that contain more than $\lceil n/k \rceil$ users and place them in the next best cluster. However, also, we have to consider users who least belong in an existing cluster, thereby reducing the increase in IL.

To achieve this, let us start from the first element in the ascending-ordered distance array, D_{i_1,j_1}, and assign user u_{i_1} to cluster C_{j_1} if it satisfies the following two conditions.

1) Size of cluster C_{j_1} should be less than or equal to $\lceil n/k \rceil$.
2) User u_{i_1} is not assigned.

Since all the clusters should be of equal size, the maximum number of users allowed per cluster is $\lceil n/k \rceil$. If any user cannot be assigned, we ignore and forward with the next element in the array. This process is explained in Algorithm 2.

Algorithm 2 Equi-Cardinal KMC

```
Input: N → Set of users, k → number of clusters,
    A → Set of attributes
Output: C → Culsters with assigned users
C ← k-meansClustering(N,k)
D_A ← CalculateDistance(N,C,A)
assigned ← n * 1 zero matrix
C ← k * 1 array of empty lists
for each element m in D_A do:
    u ← D_A(m, 2)
    c ← D_A(m, 3)
    if assigned(u) ≠ 0 AND length(C_c) ≤ ⌈n/k⌉ then
        C_c ← append u
        assigned(u) ← 1
    endif
endfor
```

C. Enhancing k-Anonymity by l-Diversity

k-anonymity is efficient and straightforward but suffers from its drawbacks. One of the significant disadvantages is that it is vulnerable to attribute disclosure. Numerous researchers have worked on this drawback to improvise the algorithm (see [47], [48]). Two other notable attacks were identified in [49]: homogeneity and background knowledge attack.

According to [49], an equivalence class is said to have l-diversity if there are at least l "well-represented" values for the sensitive attribute. In this method, the term "well-represented" can be interpreted in many ways. In this paper, we use entropy as the information-theoretic notion and employ the concept of entropy l-diversity. We use this metric to measure the diversity of each cluster

$$\text{Entropy}(E) = -\sum_{s \in S} p(E, s)\ \text{logp}(E, s)$$

where S is the set of sensitive attributes and $p(E, s)$ is the fraction of records in E that has sensitive value s. If the set S is divided into two sub-blocks S_a and S_b, then Entropy $(S) \geq$ min [Entropy(S_a), Entropy(S_b)]. Thus, to achieve l-diversity, we need to maintain an entropy of at least log(l) for the entire table. The process of this enhancement is explained in detail in Algorithm 3.

Algorithm 3 Post-Process to Ensure l-Diversity

```
Input: C → Culsters ensuring k-anonymity, l → desired
    diversity
Output: C → Culsters ensuring k-anonymity and l-
    diversity
for each cluster c in C do:
    D[c] ← diversity of cluster 'c'
endfor
remove all elements from D whose value ≥ l
while any value in D < l do
    H ← cluster with high diversity value
    L ← cluster with least diversity value
    M ← H + L
    C ← C − {H, L} + M
endwhile
```

In this algorithm, we first compute the diversity of each cluster and save them in an array called "D." However, any cluster that is at least l-diverse does not have to go through the post-processing step. Hence, we remove such clusters from our processing algorithm. The remaining clusters are then sent to the processing. Since this is a greedy algorithm, we combine the most, and the least diverse clusters to verify the combined cluster is l-diverse. Once we combine those two clusters, there are two outcomes: formed new cluster is "l"-diverse or it is not. If it is l-diverse, it is removed from the processing step and continue with the remaining clusters. If the new cluster still is not "l"-diverse, we now have new clusters in the processing algorithm and continue with our step. We keep doing this until all the clusters are at least "l"-diverse. In all of this process, since we are not removing any elements from the cluster, our k-anonymity principle still holds for the newly formed clusters.

D. Edge Anonymization

It is also important to observe that the information leak can happen through edges. By observing how the nodes are connected in a graph, an attacker can gain insight on whose that node might be if the related background information is provided [50]–[53]. Hence, it is imperative to conceal the link information between nodes. Our clustering technique will group users with similarity. Now, all the users in a single group

are represented by a single cluster head. Hence, the edges between the users belonging to two clusters can be modified into super edges. Let us say there are two users u_1 and u_2 in cluster c_1 and two users u_3 and u_4 in cluster c_2. Our initial OSN contains edges between the users $u_1 \rightarrow u_2$, $u_1 \rightarrow u_3$, and $u_2 \rightarrow u_4$. Therefore, there are two edges between clusters c_1 and c_2 and one edge inside cluster c_1. Since we are clustering all the similar users into a single cluster; we ignore all the edges between them. However, we have to focus on inter-cluster edges. Since there are two such edges, we now have a super edge with weight 2. Also, to impede more information from the attacker; we neutralize the weight by averaging with the number of users in both clusters. The final super edge between the clusters c_m and c_n can be calculated as

$$e_{c_m,c_n} = \frac{\sum e_{u_{c_m},u_{c_n}}}{|c_m| + |c_n|} \tag{3}$$

where $e_{u_{c_m},u_{c_n}}$ is an edge, where one node of the edge belongs to user in cluster c_m and the other belongs to user in cluster c_n, and $|c_i|$ is the number of users in cluster c_i.

E. Weighted Directed Graph

A real-world social network has edges that are weighted and directed. Many social networks, such as Twitter, Instagram, and Reserchgate, have a concept called followers and followed. Every user has some followers, and he/she might be following other people. This concept raises the need of direction for the edges. A celebrity with 1000 followers and following a single person is different than a regular person who follows 1000 celebrities and is followed by a single person. Thus, we give an outward directed edge for the concept "follows" and an inward directed edge for the concept "followed by." In the previous example, the celebrity has 100 inward directed edges and one outward directed edge, while the regular person has 100 outward directed edges and one inward directed edge.

For such social networks, ignoring the direction of edges results in huge IL. Hence, we propose to maintain two different edges between the clusters. There will be two edges between any two given clusters having both outward-directed and inward-directed edges. The edge weights of both the edges are calculated using the edge anonymization technique discussed in Section IV-D. This process is explained in detail in Algorithm 4.

Algorithm 4 Edge Anonymization for a Weighted Directed Network

```
Input: C → Culsters ensuring k-anonymity and l-diversity,
       E → original edge set of all users
Output: C → Culsters ensuring k-anonymity and l-
       diversity and are edge anonymized
for each cluster pair c, c' in C do:
    n → number of users in cluster c
    m → number of users in cluster c'
    outward = 0
    inward = 0
    for each outward edge e_out from c to c' do:
        outward = outward + e_out
    endfor
    for each inward edge e_in in C from c to c' do:
        inward = inward + e_in
    endfor
    Outward edge weight from c to c' → outward/|m+n|
    Inward edge weight from c to c' → inward/|m+n|
endfor
```

V. Preservation

To evaluate the proposed algorithm, we intend to measure its performance based on two metrics: IL and degree of anonymization (DA).

A. Degree of Anonymization

According to k-anonymity [28], a graph is k-anonymous if for every node "v," there exist at least $k-1$ other nodes in the graph with the same degree as "v," that is, the user degree is the same as the degree of its assigned cluster. There are $\lfloor n/(k-1) \rfloor$ more users with the same degree. Hence, we say that by assigning the users to clusters with the algorithm mentioned in Algorithm 2, we obtain an (n/k) degree anonymous graph. We calculate DA as

$$\text{DA} = \text{Degree}(C_{u_i}) * l. \tag{4}$$

B. Information Loss

According to [54], IL can be calculated as

$$\text{Information Loss}(L) = \frac{\text{SSE}}{\text{SST}} \tag{5}$$

where SSE is the sum of squares within group and SST is the sum of squares between the groups

$$\text{SSE} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (x_{ij} - \overline{C_l})'(x_{ij} - \overline{C_l}) \tag{6}$$

where K is the number of clusters, n_i is the user n that belongs to cluster C_i, and $\overline{C_i}$ is the average data vector over the whole set of users belonging to cluster C_i

$$\text{SSA} = \sum_{i=1}^{k} \frac{t_i^2}{n_i} - \frac{t^2}{n} \tag{7}$$

where t_i is the total sum of users in cluster C_i, T is the grand total of all clusters, n_i is the user n that belongs to cluster C_i, and n is the total number of users.

Now, let us analyze the privacy preservation of the proposed mechanism.

Theorem 1: For each node u, the probability that an attacker re-identifies "u" lies in the range $(\lfloor n/k^2 \rfloor, \lceil n/k \rceil)$.

Proof: Let us consider the initial graph or the un-anonymized graph $G = (V, E)$, where V is the user set, E is the edge set, and $|v| = n$. Let us assume the anonymized graph as $G = (K, E)$, where K is the number of clusters and E is the modified edge set. Also, according to the proposed algorithm, each cluster has at least $\lfloor n/k \rfloor$ users. Let us assume that the attacker has background knowledge of a user u_i, which

means that $\overline{|u_i|}$ is known. From the anonymized graph G, we know each cluster and its corresponding vector. Thus, an attacker can identify the cluster a user belongs to with the following probability:

$$p = \begin{cases} 1, & \text{if } u_i \in C_j \\ \dfrac{1}{k}, & \text{otherwise.} \end{cases} \tag{8}$$

If an attacker correctly identifies the cluster, there are still $\lfloor n/k \rfloor$ or $\lceil n/k \rceil$ users in C_j, and hence, the probability that the attacker identifies the right user is

$$p = \begin{cases} \left(\left\lfloor \dfrac{n}{k} \right\rfloor, \left\lceil \dfrac{n}{k} \right\rceil \right), & \text{if } u_i \in C_j \\ \left(\dfrac{1}{k} \left\lfloor \dfrac{n}{k} \right\rfloor, \dfrac{1}{k} \left\lceil \dfrac{n}{k} \right\rceil \right), & \text{otherwise.} \end{cases} \tag{9}$$

Hence, the range can be guaranteed as $((1/k)\lfloor n/k \rfloor, \lceil n/k \rceil) \approx (\lfloor n/k^2 \rfloor, \lceil n/k \rceil)$. □

Lemma 1: Objective 1 holds.

Proof: For a given constant "c," choose the number of clusters "k" in such a way that

$$\frac{1}{c} \geq \frac{n}{k^2}. \tag{10}$$

This can be further reduced as

$$k = \lceil \sqrt{cn} \rceil. \tag{11}$$

□

Theorem 2: For a given set of users (n) and defined number of clusters (k), the minimum anonymity degree for any user u_i in the network is

$$D(u_i) = n - (k-1) * \left\lceil \frac{n}{k} \right\rceil. \tag{12}$$

Proof: According to the proposed algorithm, the distance between user 'i' and cluster centroids are

$$D(u_i, c_a) \leq D(u_i, c_b) \leq D(u_i, c_c) \leq \cdots \leq D(u_i, c_k). \tag{13}$$

By (13), it is clear that user u_i is the closest to cluster center c_a, and hence, it should be assigned to cluster c_a. However, let us assume that cluster c_a already contains $\lceil n/k \rceil$ users. Then, u_i will be assigned to the next best option, that is, cluster c_b. Therefore

$$\text{maximum number of users per cluster} = \left\lceil \frac{n}{k} \right\rceil. \tag{14}$$

Let us assume the worst case scenario of the ordered distance matrix D_A, that is, all the users are nearest to the single cluster centroid. Also, assume that $\gamma = \lceil n/k \rceil$. Then

$$\begin{pmatrix} D_{i_1,j_1} \\ D_{i_2,j_1} \\ D_{i_3,j_1} \\ \vdots \\ D_{i_n,j_1} \end{pmatrix} \begin{pmatrix} D_{i_1,j_2} \\ D_{i_2,j_2} \\ D_{i_3,j_2} \\ \vdots \\ D_{i_n,j_2} \end{pmatrix} \cdots \begin{pmatrix} D_{i_1,j_k} \\ D_{i_2,j_k} \\ D_{i_3,j_k} \\ \vdots \\ D_{i_n,j_k} \end{pmatrix}. \tag{15}$$

According to the proposed algorithm, users u_1, u_2, $u_3, \ldots, u_\gamma$ will be assigned to cluster c_1, users $u_{\gamma+1}$, $u_{\gamma+2}$, $u_{\gamma+3}, \ldots, u_{2\gamma}$ will be assigned to cluster c_2, and users $u_{(n-1)(\gamma+1)}, u_{(n-1)(\gamma+2)}, u_{(n-1)(\gamma+3)}, \ldots, u_n$ will be assigned to cluster c_k.

Thus, clusters c_1–c_{n-1} have $\lceil n/k \rceil$ users. Remaining users belong to cluster c_k. Total number of users in $c_1 = \lceil n/k \rceil$. Total number of users in c_1 through $c_{k-1} = (k-1)\lceil n/k \rceil$. Remaining users $= n - (k-1)\lceil n/k \rceil$ according to [2]

$$n - (k-1)\left\lceil \frac{n}{k} \right\rceil \leq k \left\lceil \frac{n}{k} \right\rceil. \tag{16}$$

Hence, the lower bound on data anonymity is

$$D(u_i) \geq n - (k-1)\left\lceil \frac{n}{k} \right\rceil. \tag{17}$$

□

VI. Computational Analysis

Theorem 3: Dividing the nodes into "k" clusters is a subset of edge partitioning problem for the number of users $n \geq 3$, which is NP-complete.

Proof: Given a graph $G = (V; E)$, our problem is to determine whether the edge set E can be partitioned into subsets E_1, $E_2, \ldots$ in such a way that each E_i generates a subgraph of G isomorphic to the complete graph K_n on n vertices. Our main result is that the problem EP_n is NP-complete for each $n \geq 3$. From this, we can deduce that several other edge-partition problems are NP-complete. To show that EP_n is NP-complete, we can reduce our problem to 3SAT problem, which is known to be NP-complete. A set of clauses $C = \{C_1, C_2, \ldots, C_r\}$ in variables $u_1, u_2, \ldots, u_s$ are given, each clause C_i consisting of three literals, $l_{i,1}$, $l_{i,2}$, and $l_{i,3}$, where a literal $l_{i,j}$ is either a variable u_k or its negation u_k. The problem is to determine whether C is satisfiable, that is, whether there is a truth assignment to the variables that simultaneously satisfy all the clauses in C. A clause is satisfied if exactly all of its literals has value "true"

$$\sum_{j=1 \text{ to } k} l_{i,j} = 1. \tag{18}$$

Hence, any final solution should contain exactly "k" vertices, and hence, it is an edge partition problem that is NP-complete. □

Theorem 4: Computational complexity of the proposed algorithm for "n" users with "d" attributes divided into "k" clusters is $O(n(d+k))$.

Proof: Equi-cardinal KMC, as proposed in Algorithm 2, combines two algorithms and computation of its own.

1) *k-Means Clustering:* We refer [55] to compute clusters. This algorithm has a running time (RT) of $O(nd)$ with approximation ratio of $(1+\epsilon)$.
2) *Calculation:* This is proposed in Algorithm 1 that runs in $O(nk)$ time.
3) *Re-Arranging Clusters to Form Equal-Sized Clusters:* To re-arrange clusters, we compare with the array D_A that is of size $n * k$. Hence, comparing and assigning each element in D_A take $O(nk)$ time.

Thus, the total time complexity of the algorithm is

$$\begin{aligned} &= O(nd) + O(nk) + O(nk) \\ &= O(nd) + O(nk)(as O(2nk) = O(nk)) \\ &= O(nd + nk) \\ &= O(n(d+k)). \end{aligned}$$

□

VII. Experimental Results

The effectiveness of the proposed algorithm has been verified by utilizing two different real-life data sets. The first data set is the Yelp data set [56]. The recent version of this data set was released in January 2018 and has been online for the Yelp challenge. The Yelp data set is a customer review data set, where each user is connected to several other users and has the user's profile information. This data set contains a total of 1.1 million users and their reviews. We have focused on two files in this data set: users and friends. These two files gave us information on the user profile/attributes and the connection/edge information between the users. There are a total of 18 attributes that include information about a user, such as the number of reviews given, the number of user reviews, and average stars that are given.

Another data set that is used for our experiments is the Facebook data set provided by Stanford [57]. There are a total of 1 million users in this data set and 25 attributes for each user. User attributes include gender, country, language, and residence. Facebook is a social network data. However, unlike the real-world social network, we do not have many attributes associated with each user. Hence, this data set is also useful for understanding the behavior of the proposed method.

We compare our proposed algorithms with three other algorithms. The first one is the probability-based random obfuscation (PRO) introduced in [58]. In this method, random obfuscation is achieved by perturbing graph G, that is, randomly removing edges. However, the vertices are grouped if the edge has been removed. To generate a random number, the Bernoulli trial has been used. This trial generates random probability values. If the probability is greater than 0.5, the edge will not be removed. If the probability is less than 0.5, the edge is removed.

The second method is obfuscation by 2* neighborhood weighted grouping (NWG), as introduced in [23] and [24]. In this method, we start by randomly choosing a node. We parse through all the two-neighborhood nodes and obtain $(k-1)$ nodes whose weights are large. We can extend this algorithm to any neighborhood.

The third method is a simple KMC. In this method, we cluster the users based on their profiles. These three methods are compared with our ECC algorithm proposed in Section IV-B and our final l-diversity-enhanced ECC (LECC) for weighted and directed graphs proposed in Sections IV-C–IV-E.

All the experiments were conducted on Windows 10 operating system with Intel Core Duo 2.66-GHz CPU, 12-GB Memory, and MATLAB 9.2 platform. Each observation has been averaged over 50 instances. We have devised three different experimental metrics to observe the performance of the proposed method. Each experiment considers the different settings of users and attributes. Evaluation metrics are discussed in Section IV as a part of the proposed method. Three metrics need to be observed in each experiment: IL, DA, and RT.

A. Effect of the Number of Clusters

The goal of our first experiment is to observe the performance enhancement of ECC and enhanced l-diversity clustering when compared with the traditional clustering algorithm. In this set of experiments, we ensure that both the data sets have an equal number of users that is 10 000 users. Although the number of attributes in both data sets is different, we intend to observe the difference in the performance of all three algorithms. Fig. 3 shows the experimental results for the Yelp data set. Similarly, Fig. 4 shows the experimental results for the Facebook data set.

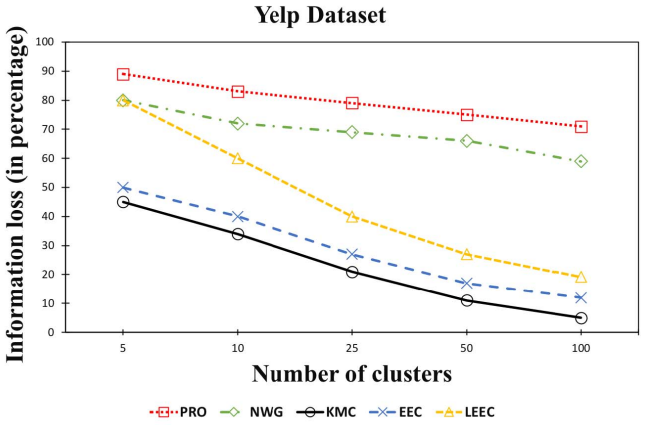

Fig. 3. IL: effect of anonymization while modifying the number of clusters for the Yelp data set.

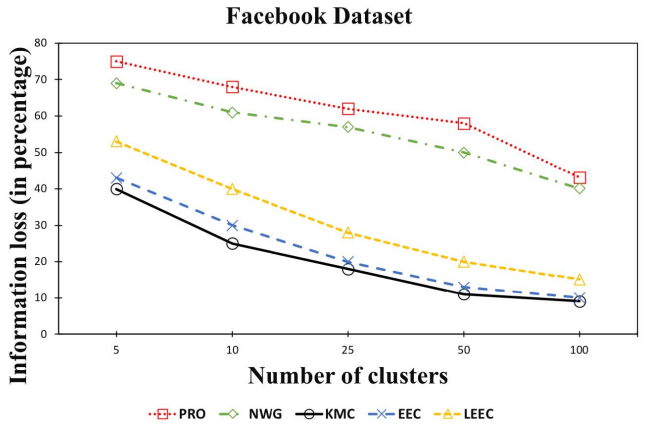

Fig. 4. IL: effect of anonymization while modifying the number of clusters for the FB data set.

We can observe from Fig. 5 that the DA has been increased at least ten times for ECC and twenty times for the l-diversity-enhanced clustering. It can also be observed that this increment in anonymization is maintained by maintaining the difference in IL to be less than 0.07% for ECC and 0.7% for the enhanced l-diversity clustering. Also, from Fig. 6, we can observe that the DA for ECC has been increased by 50 times, while the difference in IL is less than 4%. Similarly, from Fig. 6, we can observe that the anonymization has increased at almost 100 times for the enhanced l-diversity clustering. The difference between the achieved DA between the Yelp and Facebook data sets is due to the number of attributes in each data set. The Yelp data set has 18 attributes, while the Facebook data set has 25 attributes. As there is more information, clusters formed are more meaningful and closely associated. From the above-mentioned observations,

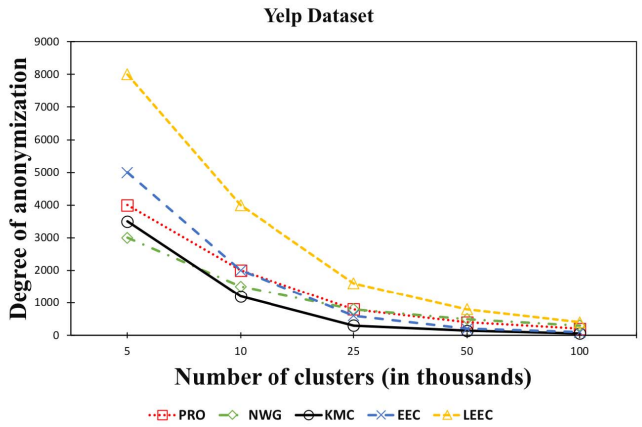

Fig. 5. IL: effect of anonymization while modifying the number of clusters for the Yelp data set.

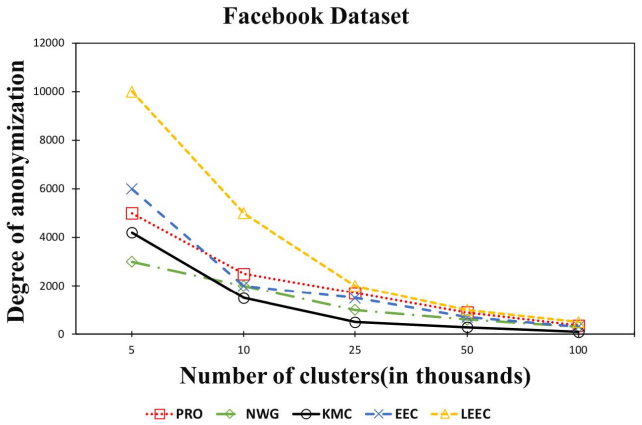

Fig. 6. DA: effect of anonymization while modifying the number of clusters for the FB data set.

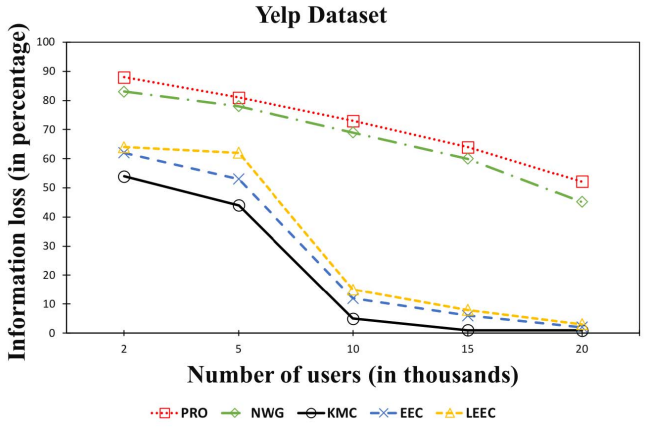

Fig. 7. IL: effect of anonymization while modifying the number of users for the Yelp data set.

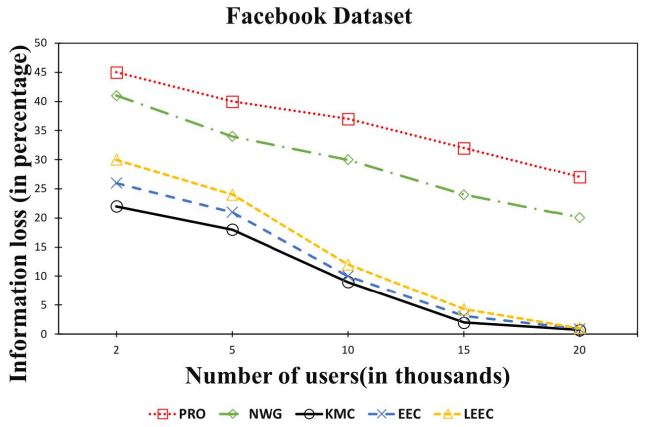

Fig. 8. IL: effect of anonymization while modifying the number of users for the FB data set.

we can see that the proposed ECC algorithm and the enhanced l-diversity clustering perform better than traditional clustering when both IL and DA are considered.

B. Effect of the Number of Users

The goal of our second experiment is to observe the performance comparison of ECC, enhanced l-diversity clustering, and traditional clustering under the influence of the number of users. Both the data sets have a huge number of user data, and we have only considered 10 000 users for our previous experiment. Clusters formed can be more meaningful if we can provide a large number of points. Hence, in this set of experiments, we vary the users from 2000 to 20 000 and observe the performance difference. Since we are focusing on a large data set, we have considered a constant number of clusters that is 100. Figs. 7 and 9 show the experimental results for the Yelp data set, while Figs. 8 and 10 show the experimental results for the Facebook data set.

From the experimental results of both the data set results in Fig. 7, it is clear that IL is decreased by increasing the number of users. This is a simple observation that, due to the increase in data points, the formed clusters are more meaningful, which means less IL. However, it is to be observed that the difference in IL between the traditional clustering and the proposed clustering algorithms is also decreasing by increasing the number of clusters. Initially, let us compare the performance enhancement of ECC over traditional clustering. From Fig. 8, it can be observed that the DA is increased with the increase in the number of users. It can also be observed that the increase in this value is almost exponential. With 2000 users, the anonymization is 6% using the Yelp data set and 5% using the Facebook data set. While with 20 000 users, the value has increased to 30% using the Yelp data set and 33% using the Facebook data set.

Our next comparison is how enhanced l-diversity clustering further increases the performance. From Fig. 8, it can be observed that the IL is 0.3% more than equi-cardinal and 0.5% more than traditional clustering at the maximum. However, it can be observed from Fig. 9 that the DA is at most two times greater than ECC. As the number of users per cluster increases, there is a huge chance of their attribute similarity.

C. Effect of the Removal of Sensitive Attributes

The goal of our third experiment is to observe the performance comparison of proposed algorithms compared to the traditional algorithm when the sensitive attributes are removed.

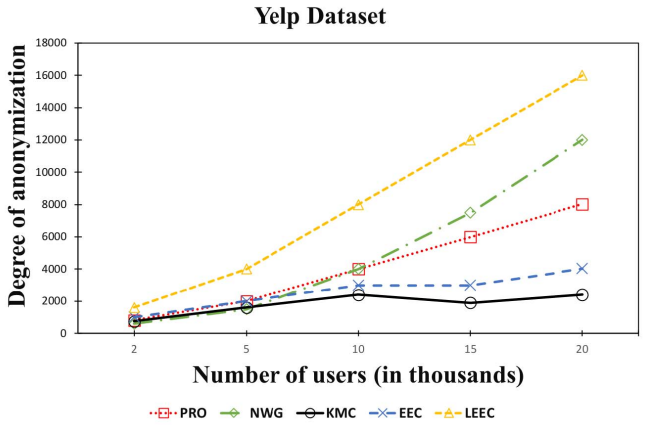

Fig. 9. DA: effect of anonymization while modifying the number of users for the Yelp data set.

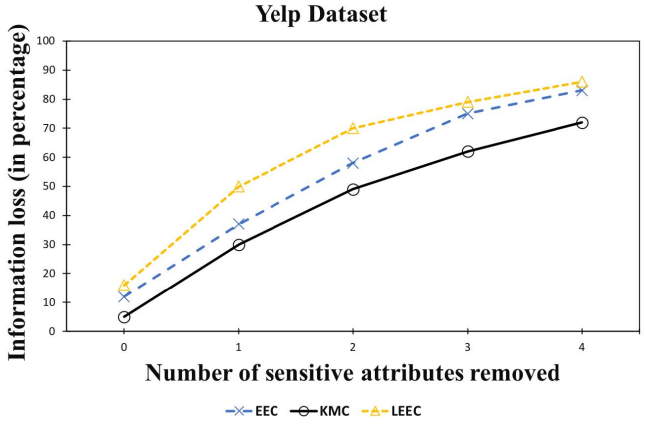

Fig. 11. IL: effect of anonymization while removing the sensitive attributes for the Yelp data set.

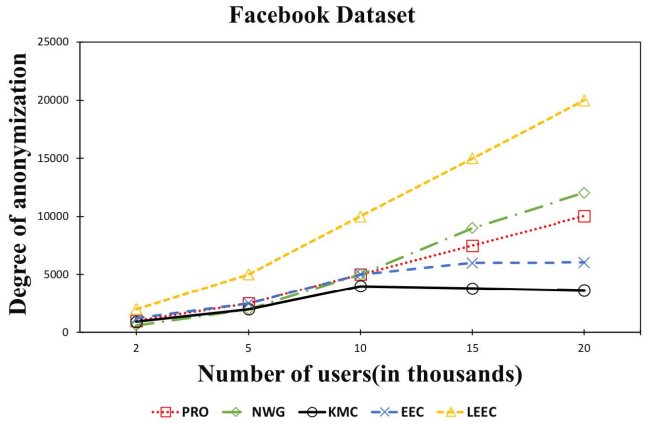

Fig. 10. DA: effect of anonymization while modifying the number of users for the FB data set.

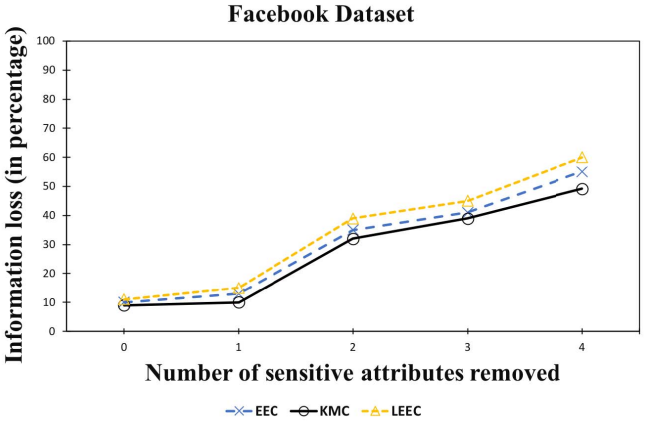

Fig. 12. IL: effect of anonymization while removing the sensitive attributes for the FB data set.

All the users in an OSN have sensitive attributes that they wish to keep private and not share publicly with third party advertisers. However, it is up to the OSN administrator to decide which attributes are sensitive. In this experiment, we consider that each user provides a list of attributes that he/she deems as sensitive. When all such data are collected from the network, the OSN administrator has an aggregate list of which set of attributes are sensitive for the entire network. From this set, he can remove the top "n" sensitive attributes. In this set of experiments, we have ranked all the attributes in the data sets according to their sensitivity toward users' privacy. We then started removing the top "n" to observe their effect on the anonymization process. Fig. 11 shows the experimental results of the Yelp data set, while Fig. 12 shows the results of the Facebook data set.

From Fig. 11, we can observe that there is a linear increment in the IL by removing the sensitive attributes. This observation can be seen in both the data sets. This observation confirms that the IL is increased irrespective of the data set. Also, the difference in IL between the naive clustering and the ECC methods has increased by 70% in the Yelp data set and by 50% in the Facebook data set. Similarly, the difference in IL between the ECC and the l-diversity-enhanced clustering method has increased by 90% in the Yelp data set and 60% in the Facebook data set. This is because as the number of attributes decreased, the intra-cluster distance between various users increases. This leads to an increase in IL in naive clustering. While the clusters formed are more sparse, removing the users from the nearest cluster head and rearranging them into other clusters further increase the IL.

It has to be noted that the DA does not change as the sensitive attributes are removed. This is because the "k" value remains the same, as there will be the same number of users in each cluster. Also, the l-value represents the average number of distinct values for each attribute. Sensitive attributes play a key role in clustering users. Hence, the values on those attributes always have minimum distinct values. By removing those sensitive attributes, we are not changing the average l-value, and hence, our DA remains the same.

VIII. Conclusion

Privacy in OSN is a significant concern, as the attacks, due to information leak, have increased in recent times. However, it is also essential for the administrators of OSN to provide anonymized information to the third-party advertisers, as it is their major source of income. Hence, there is a need to develop an anonymization technique that can anonymize the user network but not remove too much information that is

not useful for the advertisers. User anonymization and edge anonymization are current prevailing solutions, and they have their drawbacks. Hence, we have proposed a method called ECC algorithm to anonymize a social network. This method forms clusters based on user distances. We have considered user attribute as a metric for calculating the distance between users. Also, we have developed a mechanism to form the clusters of equal size, as the formed clusters ensure k-anonymity. Finally, the proposed method ensures that all the users in the network are k-anonymous, and the network is anonymized.

Experimental results have shown that the proposed method performs better than the naive clustering method when IL and DA are considered. We can achieve up to 50 times more DA with a minimal amount of increase in IL. However, other factors also contribute to the IL, such as the number of users in the network and the number of attributes associated with the users. In addition, by removing sensitive attributes, we lose information and, thus, increases the intra-cluster distance.

Future research work includes considering other graph properties to improve the algorithm further. Also, we would like to consider a dynamic social network, where users constantly change attribute values.

REFERENCES

[1] *Number of Social Media Users Worldwide 2010-17 With Forecasts to 2021 (in Billions)*. Accessed: Jun. 01, 2019. [Online]. Available: https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/

[2] N. R. Adam and J. C. Worthmann, "Security-control methods for statistical databases: A comparative study," *ACM Comput. Surv.*, vol. 21, no. 4, pp. 515–556, Dec. 1989.

[3] J. Brickell and V. Shmatikov, "The cost of privacy: Destruction of data-mining utility in anonymized data publishing," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2008, pp. 70–78.

[4] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Trans. Ind. Informat.*, to be published.

[5] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis, "Resisting structural re-identification in anonymized social networks," *Proc. VLDB Endowment*, vol. 1, no. 1, pp. 102–114, Aug. 2008.

[6] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava, "Anonymizing social networks," Dept. Comput. Sci., Univ. Massachusetts, Massachusetts, MA, USA, Tech. Rep. 19, 2007, p. 180.

[7] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart Internet of Things systems: A consideration from a privacy perspective," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 55–61, Sep. 2018.

[8] J. Cheng, A. W.-C. Fu, and J. Liu, "K-isomorphism: Privacy preserving network publication against structural attacks," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2010, pp. 459–470.

[9] E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationships in graph data," in *Privacy, Security, and Trust in KDD*, F. Bonchi, E. Ferrari, B. Malin, and Y. Saygin, Eds. Berlin, Germany: Springer, 2008, pp. 153–171.

[10] L. Zou, L. Chen, and M. T. Özsu, "K-automorphism: A general framework for privacy preserving network publication," *Proc. VLDB Endowment*, vol. 2, no. 1, pp. 946–957, Aug. 2009.

[11] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Trans. Netw. Sci. Eng.*, to be published.

[12] J. Casas-Roma, J. Herrera-Joancomartí, and V. Torra, "A survey of graph-modification techniques for privacy-preserving on networks," *Artif. Intell. Rev.*, vol. 47, no. 3, pp. 341–366, Mar. 2017.

[13] Z. Cai and Z. He, "Trading private range counting over big iot data," in *Proc. 39th IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019.

[14] 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.

[15] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Trans. Depend. Sec. Comput.*, vol. 15, no. 4, pp. 577–590, Jul./Aug. 2016.

[16] C. Dwork, "Differential privacy," *Encyclopedia Cryptogr. Secur.*, 2011, pp. 338–340.

[17] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, S. Halevi and T. Rabin, Eds. Berlin, Germany: Springer, 2006, pp. 265–284.

[18] S. Ji, P. Mittal, and R. Beyah, "Graph data anonymization, de-anonymization attacks, and de-anonymizability quantification: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1305–1326, 2nd Quart., 2016.

[19] P. Mittal, C. Papamanthou, and D. Song, "Preserving link privacy in social network based systems," 2012, *arXiv:1208.6189*. [Online]. Available: https://arxiv.org/abs/1208.6189

[20] A. Korolova, R. Motwani, S. U. Nabar, and Y. Xu, "Link privacy in social networks," in *Proc. 17th ACM Conf. Inf. Knowl. Manage.*, Oct. 2008, pp. 289–298.

[21] B. Krishnamurthy and C. E. Wills, "Characterizing privacy in online social networks," in *Proc. 1st Workshop Online Social Netw.*, Aug. 2008, pp. 37–42.

[22] A. M. Fard and K. Wang, "Neighborhood randomization for link privacy in social network analysis," *World Wide Web*, vol. 18, no. 1, pp. 9–32, Jan. 2015.

[23] Q. Liu, G. Wang, F. Li, S. Yang, and J. Wu, "Preserving privacy with probabilistic indistinguishability in weighted social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 5, pp. 1417–1429, May 2017.

[24] M. E. Skarkala, M. Maragoudakis, S. Gritzalis, L. Mitrou, H. Toivonen, and P. Moen, "Privacy preservation by k-anonymization of weighted social networks," in *Proc. Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2012, pp. 423–428.

[25] M. Siddula, L. Li, and Y. Li, "An empirical study on the privacy preservation of online social networks," *IEEE Access*, vol. 6, pp. 19912–19922, 2018.

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

[27] X. Wu, X. Ying, K. Liu, and L. Chen, *A Survey of Privacy-Preservation of Graphs and Social Networks*. Boston, MA, USA: Springer, 2010, pp. 421–453.

[28] P. Samarati, "Protecting respondents privacy in microdata release," *IEEE TransacY Tions Knowl. Data Eng.*, vol. 13, no. 6, pp. 1–18, Nov. 2001.

[29] P. Samarati, "Protecting respondents privacy in microdata release," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 6, 2001.

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

[31] B. Zhou and J. Pei, "The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks," *Knowl. Inf. Syst.*, vol. 28, no. 1, pp. 47–77, Jul. 2011.

[32] D. F. Nettleton, D. Sáez-Trumper, and V. Torra, "A comparison of two different types of online social network from a data privacy perspective," in *Proc. Int. Conf. Modeling Decis. Artif. Intell.*, Jul. 2011, pp. 223–234.

[33] 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*, May 2007, pp. 181–190.

[34] J. Goldberger and T. Tassa, "Efficient anonymizations with enhanced utility," in *Proc. IEEE Int. Conf. Data Mining Workshops*, Dec. 2009, pp. 106–113.

[35] W. Jiang and C. Clifton, "A secure distributed framework for achieving k-anonymity," *Int. J. Very Large Data Bases*, vol. 15, no. 4, pp. 316–333, Nov. 2006.

[36] M. E. Nergiz and C. Clifton, "Thoughts on k-anonymization," *Data Knowl. Eng.*, vol. 63, no. 3, pp. 622–645, Dec. 2007.

[37] S. Zhong, Z. Yang, and R. N. Wright, "Privacy-enhancing k-anonymization of customer data," in *Proc. 24th ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Syst.*, Jun. 2005, pp. 139–147.

[38] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: Privacy beyond K-anonymity," in *Proc. IEEE ICDE*, Apr. 2006, p. 24.

[39] J. Domingo-Ferrer and V. Torra, "A critique of k-anonymity and some of its enhancements," in *Proc. 3rd Int. Conf. Availability, Rel. Secur.*, Mar. 2008, pp. 990–993.

[40] J. Qian, X.-Y. Li, C. Zhang, L. Chen, T. Jung, and J. Han, "Social network de-anonymization and privacy inference with knowledge graph model," *IEEE Trans. Depend. Sec. Comput.*, vol. 16, no. 4, pp. 679–692, Jul./Aug. 2019.
[41] X. Liu and X. Yang, "A generalization based approach for anonymizing weighted social network graphs," in *Proc. 12th Int. Conf. Web-Age Inf. Manage.*, Wuhan, China. Berlin, Germany: Springer-Verlag, 2011, pp. 118–130. [Online]. Available: http://dl.acm.org/citation.cfm?id=2035562.2035578
[42] K. Liu and E. Terzi, "Towards identity anonymization on graphs," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2008, pp. 93–106.
[43] I. Kayes and A. Iamnitchi, "Privacy and security in online social networks: A survey," *Online Social Netw. Media*, vols. 3–4, pp. 1–21, Oct. 2017.
[44] D. F. Nettleton, "Information loss evaluation based on fuzzy and crisp clustering of graph statistics," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Jun. 2012, pp. 1–8.
[45] Z. He, Z. Cai, and J. Yu, "Latent-data privacy preserving with customized data utility for social network data," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 665–673, Jan. 2018.
[46] J. Zhang, Y. Yuan, X. Wang, L. Ni, J. M. Yu, and M. Zhang, "RPAR: Location privacy preserving via repartitioning anonymous region in mobile social network," *Secur. Commun. Netw.*, vol. 2018, Nov. 2018, Art. no. 6829326.
[47] T. M. Truta and B. Vinay, "Privacy protection: P-sensitive k-anonymity property," in *Proc. 22nd Int. Conf. Data Eng. Workshops (ICDEW)*, Apr. 2006, p. 94.
[48] X. Xiao and Y. Tao, "Personalized privacy preservation," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*. New York, NY, USA: ACM, Jun. 2006, pp. 229–240.
[49] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," in *Proc. 22nd Int. Conf. Data Eng. (ICDE)*, Apr. 2006, p. 24.
[50] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. Srivastava, "Class-based graph anonymization for social network data," *Proc. VLDB Endowment*, vol. 2, no. 1, pp. 766–777, 2009.
[51] A. Campan and T. M. Truta, "Data and structural k-anonymity in social networks," in *Privacy, Security, and Trust in KDD*, F. Bonchi, E. Ferrari, J. Wei, and B. Malin, Eds. Berlin, Germany: Springer-Verlag, 2009, pp. 33–54.
[52] G. Cormode, D. Srivastava, T. Yu, and Q. Zhang, "Anonymizing bipartite graph data using safe groupings," *Proc. VLDB Endowment*, vol. 1, no. 1, pp. 833–844, 2008.
[53] E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationships in graph data," in *Privacy, Security, and Trust in KDD*, F. Bonchi, E. Ferrari, B. Malin, and Y. Saygin, Eds. Berlin, Germany: Springer, 2008, pp. 153–171.
[54] J. Domingo-Ferrer and J. M. Mateo-Sanz, "Practical data-oriented microaggregation for statistical disclosure control," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 1, pp. 189–201, Jan./Feb. 2002.
[55] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 1979.
[56] *Yelp Dataset Challenge*. Accessed: Jul. 6, 2019. [Online]. Available: https://www.yelp.com/dataset/challenge
[57] *Stanford Large Network Dataset Collection*. Accessed: Jul. 1, 2019. [Online]. Available: http://snap.stanford.edu/data/
[58] F. Bonchi, A. Gionis, and T. Tassa, "Identity obfuscation in graphs through the information theoretic lens," *Inf. Sci.*, vol. 275, pp. 232–256, Aug. 2014.

Madhuri Siddula received the B.S. degree in computer science engineering from Osmania University, Hyderabad, India, and the M.S. degree in information security from the Indraprastha Institute of Information Technology, New Delhi, India. She is currently pursuing the Ph.D. degree with the Department of Computer Science, Georgia State University, Atlanta, GA, USA.

Her current research interests include social networks, privacy and security, and big data mining.

Yingshu Li received the B.S. degree from the Department of Computer Science and Engineering, Beijing Institute of Technology, Beijing, China, and the M.S. and Ph.D. degrees from the Department of Computer Science and Engineering, University of Minnesota–Twin Cities, Minneapolis, MN, USA.

She is currently an Associate Professor with the Department of Computer Science, Georgia State University, Atlanta, GA, USA. Her current research interests include wireless networking, sensor networks, sensory data management, social networks, and optimization.

Dr. Li was a recipient of the NSF CAREER Award.

Xiuzhen Cheng (F'15) 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 an Assistant Professor with the Department of Computer Science, The George Washington University, Washington, DC, USA. Her current research interests include wireless and mobile computing, sensor networks, wireless security, statistical pattern recognition, approximation algorithm design and analysis, and computational medicine.

Dr. Cheng is a member of the ACM. She received the National Science Foundation CAREER Award in 2004. She is an Editor of the *International Journal of Ad Hoc and Ubiquitous Computing* and *International Journal of Sensor Networks*.

Zhi Tian was on the Faculty of Michigan Technological University, Houghton, MI, USA, from 2000 to 2014. She served a three-year term as a Program Director at the U.S. National Science Foundation. She has been a Professor with the Electrical and Computer Engineering Department, George Mason University, Fairfax, VA, USA, since 2015. Her current research interests include 5G wireless communications, high-dimensional statistical signal processing, and decentralized optimization and learning.

Dr. Tian received the 2018 Communication Society TCCN Publication Award. She served on many posts with the IEEE, including the General Chair of the 2016 IEEE GlobalSIP Conference, the Chair of the Big Data Special Interest Group of the IEEE Signal Processing Society, and a Member-at-Large of the Board of Governors of the Signal Processing Society. She served as an Associate Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and IEEE TRANSACTIONS ON SIGNAL PROCESSING. She was the IEEE Distinguished Lecturer for the IEEE Communications Society and the IEEE Vehicular Technology Society.

Zhipeng Cai received the B.S. degree from the Beijing Institute of Technology, Beijing, China, and the M.S. and Ph.D. degrees from the Department of Computing Science, University of Alberta, Edmonton, AB, Canada.

He is currently an Associate Professor with the Department of Computer Science, Georgia State University, Atlanta, GA, USA. He has published more than 50 journals papers, including more than 20 IEEE/ACM TRANSACTIONS, such as the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE/ACM TRANSACTIONS ON NETWORKING, and IEEE TRANSACTIONS ON MOBILE COMPUTING. His current research interests include networking, privacy, and big data.