Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 4080047, 18 pages
https://doi.org/10.1155/2022/4080047

Research Article

LDPCD: A Novel Method for Locally Differentially Private Community Detection

Zhejian Zhang

College of Computer Science, Chongqing University, Chongqing 400044, China

Correspondence should be addressed to Zhejian Zhang; zzhejian@cqu.edu.cn

Received 4 November 2021; Revised 1 December 2021; Accepted 3 December 2021; Published 10 January 2022

Academic Editor: Bai Yuan Ding

Copyright © 2022 Zhejian Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As one of the cores of data analysis in large social networks, community detection has become a hot research topic in recent years. However, user's real social relationship may be at risk of privacy leakage and threatened by inference attacks because of the semitrusted server. As a result, community detection in social graphs under local differential privacy has gradually aroused the interest of industry and academia. On the one hand, the distortion of user's real data caused by existing privacy-preserving mechanisms can have a serious impact on the mining process of densely connected local graph structure, resulting in low utility of the final community division. On the other hand, private community detection requires to use the results of multiple user-server interactions to adjust user's partition, which inevitably leads to excessive allocation of privacy budget and large error of perturbed data. For these reasons, a new community detection method based on the local differential privacy model (named LDPCD) is proposed in this paper. Due to the introduction of truncated Laplace mechanism, the accuracy of user perturbation data is improved. In addition, the community divisive algorithm based on extremal optimization (EO) is also refined to reduce the number of interactions between users and the server. Thus, the total privacy overhead is reduced and strong privacy protection is guaranteed. Finally, LDPCD is applied in two commonly used real-world datasets, and its advantage is experimentally validated compared with two state-of-the-art methods.

1. Introduction

Due to the rapid development of Internet technology, APPs with various functions have brought great convenience to the daily interaction among users. After integration, these relationship data of users can be exploited to build social graphs, from which service providers can mine valuable information such as frequent subgraphs [1, 2], average path length among users [3], and the community structure of users [4–6]. In particular, community structure is an important feature of social graphs. On the one hand, based on network topology architecture and user attributes, various user communities and interest groups will be mined for the futural personalized recommendations [7, 8]. On the other hand, as an important manifestation of topological features, community structure has a significant guiding role in creating synthetic social graphs [9]. Therefore, the exploration of community information from social networks has attracted extensive attention in the field of academy and industry.

Community detection on the social graph requires the collection of users' social relationships. However, most social links among users are sensitive and private information. If the user uploads such data without reservation or the server does not take any privacy protection measures in the centralized data analysis, the user's local information may be exposed to the risk of leakage and inference attack. For example, in 2018, Facebook was accused of leaking tens of millions of user personal information to the UK-based third-party firm Cambridge Analytica [10]. This privacy scandal indicates that one of the main problems to be solved in the community mining of social networks is the privacy protection during the collection of user relationship data.

As far as we know, a promising model which can be utilized to resolve the privacy issue posed by the untrusted service provider is local differential privacy (LDP) [11]. LDP

is a privacy protection framework inherited from centralized differential privacy (DP or CDP) for privacy protection during data collection. The real data are perturbed by the user on the local terminal and then uploaded to the data curator. Both in industrial production areas and academic research studies [12–18], this model has been widely utilized because of its strong resistance to attack based on any background knowledge and its exclusion of the assumption of a fully trusted server.

At present, community detection based on LDP has become a research hotspot in privacy protection of big data [8, 19, 20]. In this context, the protective measures of social relationship data are transferred from centralized overall processing to distributed processing by each user. This data collection pattern poses a key problem for the community detection task of social graphs. Since most of the community detection algorithms of graph data require that the real and specific structure of the entire network be known, under the local privacy protection, it is difficult for the server to directly conduct community detection algorithm with false information uploaded by users instead of a true social graph. Moreover, serious damage to the network topology will be posed when the users independently add noise to their local relationship data. This will eventually cause excessive loss of graph structure information and seriously affect the utility of community detection results [19–21]. Therefore, it is speculated that studying the community detection problem of the social graph under LDP is difficult.

This paper intends to respectively modify existing community detection method of graph data and perturbation method of users' local data under LDP. Thus, a novel privacy protection community detection method is designed, which is named LDPCD. In its framework, a community divisive algorithm based on extremal optimization (EO) is used as the basis of the community detection method. When executing the EO algorithm, the total number of queries on the user's degree vector and the times of grouping adjustment will increase sharply with the network scale. Therefore, under a limited total privacy budget, there exists a problem of extremely large errors in the results of a single query. In this regard, the truncated Laplace mechanism is used to limit the disturbance range of the user's degree, thereby reducing the calculation error of the user's fitness value. Appropriate refinements are also made to the existing EO algorithm under the protection of LDP to significantly reduce the total interaction times between users and the server. Finally, LDPCD is adopted to conduct community detection on two social network datasets to prove its effectiveness. In summary, the following contributions of this paper are made:

(1)A novel community detection method LDPCD under local differential privacy protection is proposed, which can obtain better community detection results under higher local privacy protection requirements.

(2) In order to solve the problem of large error caused by Laplace mechanism, the truncated Laplace mechanism is introduced to optimize the local perturbation of user's degree vector. Moreover, we provide rigorous theoretical proof that the new noise addition method satisfies ε-LDP.

(3) By refining the community divisive algorithm based on extremal optimization, the interaction times between users and the server as well as the total privacy cost are reduced, and the utility of community division is also guaranteed.

(4) Through experimental evaluation on two commonly used social network datasets, the community detection results of LDPCD are compared with those of two state-of-the-art methods of graph data analysis under LDP [19, 20] to demonstrate the accuracy and effectiveness of our proposed method.

The rest of the paper is structured as follows. Section 2 introduces the research status of LDP in graph data analysis. Section 3 elaborates the preliminary knowledge of community detection and LDP protection on graph data and gives the definition of the problem in this paper. Section 4 describes the framework of LDPCD and its implementation details. Section 5 presents and analyzes the experimental results. Finally, Section 6 draws the conclusion.

2. Related Works

In recent years, research studies on protecting the topological characters and user relationships in social graph data under LDP have attracted widespread attention of scholars [8, 19, 20, 22–28]. According to different analysis objects of social graph data, the existing work can be summarized in two aspects, which are statistical graph metrics estimation and synthetic social graph generation.

2.1. The Application of LDP in Statistical Graph Metrics Estimation. The statistical metrics of the social graph are important objects of graph data mining. As coarse-grained information, these statistical metrics highly condense certain properties of the social graph, which can express complex topological relationships through simple numerical values. Therefore, some studies adopted LDP mechanisms to perturb user's social data and analyzed many types of statistical graph metrics such as degree distribution [8, 29], clustering coefficient distribution [20, 30], edge weight distribution [31], and modularity [20].

Jacob et al. [23] proposed a method for estimating the frequency of subgraphs based on LDP. The central server aggregates this local statistical information with calibrated noise after interacting with the users to estimate the total number of k-stars and triangles in the entire graph. Sun et al. [25] formulated a stringent definition of decentralized differential privacy to provide adequate protection for the information of each user and her neighbors. The total frequency of triangles, three-hop paths, and k-clique in the social graph is thus precisely estimated using a noise injection method that satisfies this privacy definition. Wei et al. [8] proposed using LDP in the collection of attribute graph data. After the random-jump perturbation to the user's

degree and the randomized response mechanism to the user's binary attribute value, the original degree distribution and the joint distribution of attribute data are, respectively, restored by unbiased estimation and EM algorithm. For the social graph with edge attributes, Liu et al. [24] proposed a novel privacy definition (attribute-wise LDP) with stronger protection than edge differential privacy. Accordingly, a novel perturbation mechanism was designed to protect all edges with the same attribute for each user. The corresponding method for the restoration of statistical metrics was also proposed to estimate the frequency of nodes with certain attribute edges and the degree-attribute joint distribution of the social graph. Ye et al. [20, 27] argued that, for the estimation of most statistical graph metrics (such as node clustering coefficient and subgraph modularity) under LDP, it is sufficient to only query the noisy degree and adjacent bit vector of each user. Based on this point, they proposed a general framework LFGDPR, which analyzes the optimal allocation scheme of privacy budget to separately perturb the two items and provides the corresponding unbiased estimation algorithm for different graph metrics.

2.2. The Application of LDP in Synthetic Social Graph Generation. In the various applications of differential privacy on social graph data, it is a popular but challenging research task to use appropriate graph generation models to generate a synthetic and privacy-guaranteed social graph for its publishing to third parties [22, 32]. With the rise of social graph analysis based on LDP, the research on the synthesis of a private graph based on the user's decentralized perturbed information is also gradually unfolding [8, 19, 28].

Qin et al. [19] conducted pioneering research on this field and proposed a graph data collection and synthetic graph generation method under LDP, which is named LDPGen. In more detail, each time LDPGen partitions all users into disjoint groups and queries each user for her perturbed degrees under the grouping, users with similar degree vectors are clustered together to form a new user partition. This process of grouping-inquiry-grouping iterates until the privacy budget is depleted. Based on the final partition, a synthetic social graph is generated by using the graph-generating model of Chung-Lu [33] for further analysis. With the same research object as in [19], Zhang et al. [28] proposed to collect users' noisy degrees by means of secure multiparty computation to form several user groups. Then, each user adopts an optimized randomized response scheme to perturb its adjacency vectors in different groups. Finally, the synthetic social graph is generated through the synthesized adjacency matrix after aggregating the perturbed bit vectors of all users. Based on the collected noisy data from all users, Wei et al. [8] used the attribute graph model (AGM) [34] and takes the estimated distribution of degrees and attribute values as input parameters to generate the initial seed graph. In order to preserve the structure and community information of the original graph, the community detection algorithm of CESNA [6] is adopted. Through continuous iterative community detection of the seed graph and the modification of the edges and attribute values in it, the convergent synthetic attribute graph with high utility is finally generated. In addition, Ye et al. [20] proposed the LFGDPR framework that can be applied to the unbiased estimation of the modularity of any subgraph. On this basis, the Louvain community detection algorithm [4] is used to divide users into communities, and a new synthetic social graph is generated based on the community detection results.

In general, the existing work can analyze some commonly used statistical metrics of graph data and generate synthetic social graphs under LDP. However, in some of it, the problem of community detection is mostly presented as a part of the whole research content and closely connected with the final synthetic graphs, which means that the quality of the graphs will have a significant impact on the utility of the community division result. In this paper, we attempt to design a more straightforward method without private graph generation and gradually restore the community structure through multiple user-server interactions.

3. Problem Definition

In this section, we briefly introduce the prerequisite knowledge of community detection and local differential privacy for graph data; then, we give a detailed definition of the considered problem. Table 1 describes the meaning of some notations used in this study.

3.1. Nonprivate Community Detection in Social Graphs. Many research studies have been conducted on the community detection of social graphs. Most classical methods of community detection are dedicated to optimizing the modularity of the community division of the entire graph. Among them, the EO-based heuristic algorithm has been widely used owing to its high computational efficiency and fast convergence speed [5].

In the method of EO-based community detection, the global variable is the modularity Q of a community division of all users, whereas the local variables are the contribution of each node to the total modularity. The contribution q_i of node i is as follows:

$$q_i = \delta_{c(i)} - \delta_i a_{c(i)}, \tag{1}$$

where $\delta_{c(i)}$ represents the number of edges connected between a node i belonging to community c and the other nodes in the same community, δ_i represents the total degree of node i, and $a_{c(i)}$ represents the proportion of the degree sum of all nodes in community c to the degree sum of all nodes in the entire network. The relationship between the modularity Q as a global variable and the local variable q_i is as follows:

$$Q = \frac{1}{2L}\sum_i q_i = \sum_c \left[\frac{L_c}{L} - \left(\frac{\sum_{i\in c}\delta_i}{2L}\right)^2\right], \tag{2}$$

where L_c and L represent the number of edges in community c and the total edges in the entire network, respectively. Since the value range of Q is $[-1/2, 1]$, to maintain the

Table 1: The description of the main notations used in this article.

Symbols	Descriptions
$\mathbf{U}_0$	The set of all users
$\mathbf{U}$	A user subset of $\mathbf{U}_0$
$\mathbf{G}$	A bipartition of $\mathbf{U}$
$\lambda_i/\widetilde{\lambda}_i$	The true/noisy fitness of user i in terms of $\mathbf{G}$
$\delta_i/\widetilde{\delta}_i$	i's true/noisy degree in $\mathbf{G}$
$\boldsymbol{\delta}_i/\widetilde{\boldsymbol{\delta}}_i$	i's true/noisy degree vector in $\mathbf{G}$
$\widetilde{Q}$	The estimated modularity of a community division of all users
$\widetilde{Q}_{\mathrm{b}}$	The estimated bipartition modularity of $\mathbf{G}$
$\mathbf{G}_{\mathrm{f}}$	The final bipartite grouping of $\mathbf{U}$ with converged $\widetilde{Q}_{\mathrm{b}}$
$\mathbf{C}_r$	The community division result of all users of the rth round bipartition
$\mathbf{U}_r$	A certain user subset (community) of $\mathbf{C}_r$
$\Delta\widetilde{Q}$	The gain of $\widetilde{Q}$ caused by the substitution of $\mathbf{G}_{\mathrm{f}}$ for $\{\mathbf{U}\}$ in user community division
ε	The privacy budget used for each query on user's degree vector based on $\mathbf{G}$
ε_{f}	The privacy budget used for the query on user's degree vector based on $\mathbf{G}_{\mathrm{f}}$ to estimate $\Delta\widetilde{Q}$

consistency, the local variable is normalized as λ_i with the same range, which is defined as the fitness of user i:

$$\lambda_i = \frac{\delta_{c(i)}}{\delta_i} - a_{c(i)}. \tag{3}$$

Therefore, the greater the fitness value of a node, the greater its contribution to the modularity of the community structure.

After defining the fitness of each user, the heuristic divisive algorithm of EO can be described as follows:

(1) Initialization: the entire network is randomly divided into two groups, each of which has the same number of nodes. This is regarded as the initial community structure of the network.

(2) Iteration: in each iteration, after the fitness values of all users have been calculated and sorted, the node with the lowest fitness is considered to contribute the least to the modularity of current bipartition of users and is moved to the other group. After each move, calculating the new two-dimensional degree vector of all users based on the original graph and updating their fitness accordingly is necessary.

By repeating step (2), an optimal bipartition state will be finally obtained. In addition, its modularity Q_{b} (also defined as bipartition modularity) reaches a locally optimal value and no longer increases. Afterward, all edges between the two resulting groups are removed, and the abovementioned initialization grouping and iteration process is independently continued in each subgraph formed by the final groups (each with their own Q_{b} when divided into two parts), thereby further splitting the users' community.

3.2. Threat Model. The community detection algorithm described in this paper involves multiple rounds of interaction between two participants, i.e., the user and the server. The user is considered to be trusted because she only keeps her social relationship data locally. However, the server is considered semitrusted. On the one hand, the central server collects true relationship data uploaded by users and reconstructs the real social graph to provide users with personalized services based on the mining results of it. On the other hand, the users' real data may be disclosed to other untrusted third parties. In addition, even if a user only uploads the true statistical values of some coarse-grained information (such as user's degree), the central server will infer whether there is a social link between this user and another targeted user based on his existing background knowledge or true information provided from other users in collusion.

3.3. Privacy Definition. Based on the threat model described in Section 3.2, to collect user's private social relationships without relying on a trusted server, we should resort to LDP mechanisms for the protection of each user's local and limited information.

In the graph data analysis of LDP, privacy definitions are generally divided into the node LDP and edge LDP [19]. In our scenario, whether two targeted users have a friendly relationship is considered private information to be protected, which coincides with the definition of edge LDP. Considering that any edge in a social graph can affect at most one bit of each user's neighbor list, edge LDP is defined as follows:

Definition 1 (edge LDP, see [19]). For a social network with N user nodes, a randomized mechanism M defined on $\{0,1\}^N$ satisfies ε-edge LDP if any user i and her two neighbor lists l_i, l_i' differ only in one bit, as well as any possible output subset $S \in \mathrm{Range}(M)$. The following probability inequality holds:

$$\Pr[M(l_i) \in S] \leq e^{\varepsilon} \Pr[M(l_i') \in S]. \tag{4}$$

Regarding the EO-based community detection algorithm adopted in this paper, the calculation of user fitness mainly involves the user's degree vector for a certain grouping situation. Thus, we adopt the degree perturbation mechanism proposed in [19]. In particular, the central server divides all users into k groups, denoted as $\xi = \{U_1, \ldots, U_k\}$. After the grouping information is distributed to all users, each user tallies her degree in each group and obtains the corresponding degree vector $\boldsymbol{\delta}_i = \{\delta_i^1, \ldots, \delta_i^k\}$. Because the

presence or absence of an edge in the social graph will affect at most one degree value in the degree vector of each user by 1, the user adds independent Laplace noise with a mean value of 0 and a scaling parameter of $1/\varepsilon$ to each dimension of the degree vector, i.e.,

$$\widetilde{\boldsymbol{\delta}}_i = \left\{\widetilde{\delta}_i^1, \ldots, \widetilde{\delta}_i^k\right\} = \left\{\delta_i^1 + \mathrm{Lap}\left(\frac{1}{\varepsilon}\right), \ldots, \delta_i^k + \mathrm{Lap}\left(\frac{1}{\varepsilon}\right)\right\}. \quad (5)$$

Therefore, if the neighboring degree vectors $\boldsymbol{\delta}_i$ and $\boldsymbol{\delta}_j$ satisfy $\delta_i^r = \delta_j^r$ for $r \in [1, k]$ and $r \neq m$, as well as $|\delta_i^m - \delta_j^m| = 1$, then, for any possible output result $\mathbf{s} = (s_1, \ldots, s_k) \in \mathrm{Range}(M)$, there is

$$\begin{aligned}\frac{\Pr[M(\boldsymbol{\delta}_i) = \mathbf{s}]}{\Pr[M(\boldsymbol{\delta}_j) = \mathbf{s}]} &= \frac{\Pr\left[\widetilde{\delta}_i^1 = s_1\right], \ldots, \Pr\left[\widetilde{\delta}_i^k = s_k\right]}{\Pr\left[\widetilde{\delta}_j^1 = s_1\right], \ldots, \Pr\left[\widetilde{\delta}_j^k = s_k\right]} \\ &= \frac{\Pr\left[\widetilde{\delta}_i^m = s_m\right]}{\Pr\left[\widetilde{\delta}_j^m = s_m\right]} \leq e^{\varepsilon}.\end{aligned} \quad (6)$$

This shows that the degree vector perturbation mechanism satisfies ε-edge LDP.

In addition, considering our community detection scenario, the user and the server interact several times in the iterative process of fitness calculation and grouping adjustment. Since any relationship edge of one user affects the degree results for each query of the server to the greatest extent, according to the sequential composition property of DP [35], in the entire process of community detection, the total privacy cost for each user is equal to the sum of all the privacy budget consumed by her interactions with the server.

3.4. Accuracy Definition. The community detection on a real social network can get the community division of users close to the facts. In this study, we use the community detection result of a real social network based on the EO algorithm as the ground truth, denoted by $\mathbf{C}_t$. The result of community detection under LDP is expressed as $\mathbf{C}_p$. For a certain deviation between $\mathbf{C}_p$ and $\mathbf{C}_t$, the accuracy of privacy-preserving community detection is measured using three metrics: modularity, ARI, and AMI. These three metrics are described in detail in Section 5.1.3.

3.5. Problem Statement. In this study, we aim to find a tradeoff between the privacy protection of user social relationship data and the utility of community detection results. Based on the definitions of privacy and accuracy given in Sections 3.3 and 3.4, as well as the EO-based community detection method, in this section, we formally describe the problem of the community detection for social networks under LDP guarantees.

Definition 2 (community detection for social networks under LDP). In this study, LDPCD, a novel framework, is designed for community detection on social networks under LDP.

(1) The framework guarantees that any relationship data of each user satisfies ε-differential privacy for the other users and the server

(2) Based on the EO algorithm, the framework exploits the degree vector information uploaded by users and uses a community divisive algorithm to divide users into several communities

(3) The framework improves the existing classical data perturbation mechanism to ensure that the results of community detection can achieve reasonable utility under strong privacy protection

Studying community detection under LDP protection is challenging because of two main reasons. First, the central server cannot use agglomerative algorithms to cluster users into communities, which is because these algorithms cannot be implemented by correctly inferring the truthfulness of the perturbed links in the local region with high probability, thereby affecting the fusion process of nodes and communities based on the greedy algorithm. Second, in the context of LDP, each user interacts with the server several times to continuously optimize the community division results. For a moderate total privacy budget, overly small distributed budget may result in the utility of community division to not significantly improve with the increase in the number of interactions. Therefore, in response to the first one, the community divisive algorithm is used as our basic method, and user statistics such as the degree vector are exploited to guide the division and adjustment of user communities. To mitigate the large perturbation error caused by excessive interactions, the conventional data perturbation mechanism is improved to enhance the accuracy of community detection under reasonable privacy protection strength.

4. Methods

In this section, we propose LDPCD, a novel framework, for community detection of social graphs under LDP protection. First, this framework is described in detail. Then, the drawbacks of directly applying the existing noise injection method of LDP to the EO-based community detection algorithm are analyzed. Finally, a modified data perturbation satisfying LDP and a refined EO-based community detection algorithm are proposed.

4.1. Framework. As shown in Figure 1, LDPCD mainly comprises two building blocks: the multiple interactions between users and the server and the server-side iterative processes of the generation of user communities. The user-side operation mainly comprises the calculation and local perturbation of the degree vector. The iterative processes comprise the multiple degree queries and iterative adjustment of user's bipartition until the bipartition modularity converges, and the iterative optimal bipartition of user's community until the user partition is stable.

At the user side, user i receives the bipartite grouping $\mathbf{G}$ from the server, generates her true degree vector $\boldsymbol{\delta}_i$, and

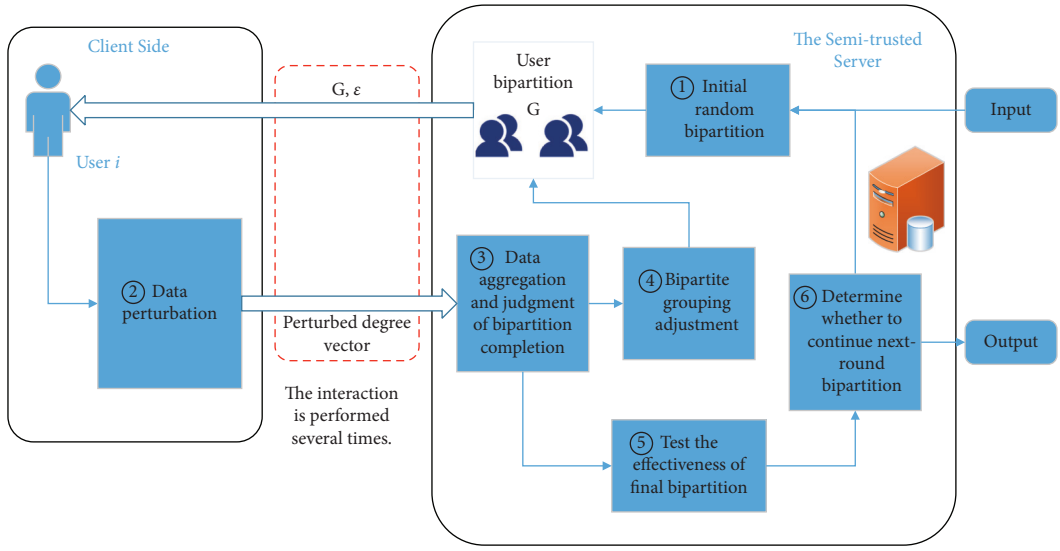


Figure 1: The framework of LDPCD.

perturbs it according to the assigned privacy budget ε and then uploads the perturbed degree vector $\tilde{\boldsymbol{\delta}}_{\mathbf{i}}$ (step ②).

At the server side, the overall data operation process is shown in steps ① and ③ to ⑥ in Figure 1. In the input step, the server gets the ID information of all surveyed users and forms a user set $\mathbf{U}_0$. Meanwhile, a community division $\mathbf{C}_0$ is initialized, where $\mathbf{C}_0 = \{\mathbf{U}_0\}$. Next, the server performs multiple rounds of bipartition to divide all users into several communities. In particular, when the bipartition process reaches the rth round, the server starts from the user community division result of the previous round and initializes the rth round's community as $\mathbf{C}_r = \varnothing$, then sets each subset in $\mathbf{C}_{r-1}$ as an independent user community $\mathbf{U}$, and performs the bipartition operation separately.

In the initial case of the bipartition operation, the server side randomly bisects $\mathbf{U}$ and obtains the initial bipartite grouping $\mathbf{G} = \{\mathbf{U}^1, \mathbf{U}^2\}$ (step ①). After querying the perturbed degree vectors of all users (step ②), the server collects them and computes the corresponding fitness for each user in the EO-based algorithm and the bipartition modularity value of the current grouping (step ③). Afterward, the server iteratively sorts the users' fitness to obtain the sequence $\tilde{\boldsymbol{\lambda}}$, adjusts the bipartite grouping according to $\tilde{\boldsymbol{\lambda}}$, and recalculates the fitness of each user until the grouping situation stabilizes (step ④). The stabilized $\mathbf{G}$ and privacy budget ε are distributed by the server to each user in $\mathbf{U}$ for the next query regarding degree vectors (step ②). After collecting all perturbed data, the server repeats the computation of fitness as well as bipartition modularity and the iterative adjustment of the user's grouping. The adjustment-query-adjustment process for the final bipartition result will last until the bipartition modularities obtained by two neighboring queries have a subtle gap, which indicates that the optimal bipartition of $\mathbf{U}$ is finally obtained and its final grouping $\mathbf{G}_f$ is thus formed (step ③).

During the rth round of user community division, after judging that the optimal bipartition of each subset $\mathbf{U}_{r-1} \in \mathbf{C}_{r-1}$ is completed (step ③), checking whether its $\mathbf{G}_f$ can cause an increase in the total modularity $\tilde{Q}$ compared to itself without bipartition (step ⑤) is necessary. If the gain is greater than a certain expected error, $\mathbf{C}_r = \mathbf{C}_r \cup \mathbf{G}_f$ is computed to update the rth round's community division. Otherwise, $\mathbf{C}_r = \mathbf{C}_r \cup \{\mathbf{U}_{r-1}\}$ is executed, indicating that any bipartition cannot pose an obvious gain in $\tilde{Q}$. Whether to perform a new round of community bipartition is decided based on the comparison result of $\mathbf{C}_{r-1}$ with $\mathbf{C}_r$ (step ⑥). Thus, in the entire process of the community divisive algorithm based on EO, the server side iteratively performs the optimal bipartition of user subsets (also as communities) for several rounds, which start from the first round of community bipartition on all users and finally ends when the user community divisions of two neighboring rounds are identical, and the total modularity stabilizes to get the final community detection result of all surveyed users.

4.2. A Naive Method. Since Laplace mechanism is commonly used for differential privacy protection of numerical data, an intuitive approach is to inject Laplace noise in user's true degree vector (as mentioned in Section 3.3). After making minor adjustments to user groups based on user perturbation data, the server needs to inform all users within the bipartite grouping about the migrated users, thus making all users update their degree vectors according to the new grouping situation. On this basis, we refer to the classical EO algorithm [5] in our naive method and only move the user with the lowest fitness to the other group in each grouping adjustment and continue the query for the user's updated perturbed degree vectors. As we can imagine, the migration of one user only results in a slight change in the degree vectors of some users. However, determining whether and to what extent the degree vector of the other nodes that have not been moved have changed is impossible for the server because he has no access to the true social graph of the surveyed users under LDP.

Therefore, each time of user grouping adjustment has to be allocated some privacy budget for querying about the updated degree vectors.

4.2.1. Problems of the Naive Method. In the abovementioned naive method, the conventional Laplace mechanism is used for data perturbation. The server executes the iterative process of the classical EO algorithm by multiple interactions with users and using the perturbed data for fitness calculation and node migration. Two main problems are encountered in this process. On the one hand, with excessively small privacy budget distributed for each query, the addition of Laplace noise can considerably distort the true degree vector when there is no limitation of output range for the noisy degree. The perturbed data with large error will considerably affect the grouping adjustment process, resulting in low modularity and poor utility of the subsequent community division results. On the other hand, the classical EO algorithm requires that every time a single user node changes its group, the information of each user is requeried, which significantly increases communication cost between users and the server. Therefore, applying this naive approach to practical community detection under LDP protection is nearly impossible.

4.3. Data Perturbation. Considering the limitations of Laplace mechanism when applied to the EO-based community detection algorithm, the truncated Laplace mechanism [36] which restricts the perturbation range of user's degree is used as our data noising scheme, thereby resolving the problem of the poor utility of the community division result caused by the large error under low privacy budget.

In the Laplace distribution $f(x) = e^{-|x-\mu|/\sigma}/2\sigma$, the output range is the real domain. However, to ensure that the result of user degree is meaningful after perturbation, the output range of the LDP-based mechanism should be limited from 0 to the total number of users or to an even shorter interval of the degree value, which can prevent a small degree from being perturbed under the conventional Laplace mechanism and resulting in a negative one with large absolute value and seriously affecting the estimation accuracy of user fitness. In this regard, the truncated Laplace mechanism [36], as an improved method of Laplace mechanism, truncates the infinite range of perturbation results. Moreover, to ensure that the integral of the output probability density of all values in the truncated range is equal to 1, the probability density function $f(x)$ is multiplied by a normalization parameter, which guarantees that the true degree results are always output with the maximum probability and meanwhile increases the output probability of all values within the output interval. The integral of probability density function in this interval is 1, whereas the output probability of all values outside this truncated range is 0, thus improving data accuracy after perturbation.

As shown in Figure 2, when the users use the truncated Laplace mechanism to perform degree perturbation locally, the limit output range should be determined according to the true value of the user, and it is expressed by (L, R), where

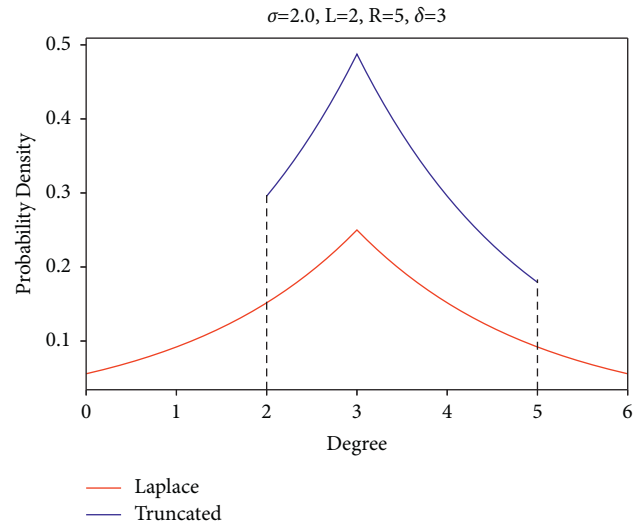


Figure 2: Laplace distribution and truncated Laplace distribution.

$L, R \in \mathbb{N}$ (here, the symbols L and R, respectively, denote the left and right boundary of the output range, and the true degree δ satisfies $L < \delta < R$). Because the information that users need to protect is the existence or absence of an edge in the true social graph, the sensitivity of each dimension of the local degree vector is $\Delta\delta = 1$. Given the privacy budget ε of each time of query, the scaling parameter σ in the truncated Laplace distribution function is obtained, i.e.,

$$\sigma = \frac{2\Delta\delta}{\varepsilon} = \frac{2}{\varepsilon}. \tag{7}$$

(In Section 4.5.1, we give detailed proof of the value of σ). Then, the integral on the intervals $(-\infty, L)$ and $(R, +\infty)$ is calculated according to the probability density function of the Laplace distribution, i.e.,

$$I_L = \int_{-\infty}^{L} \frac{1}{2\sigma} e^{-|x-\delta|/\sigma} \mathrm{d}x = \frac{1}{2} e^{-|\delta - L|/\sigma}, \tag{8}$$

$$I_R = \int_{R}^{+\infty} \frac{1}{2\sigma} e^{-|x-\delta|/\sigma} \mathrm{d}x = \frac{1}{2} e^{-|\delta - R|/\sigma}. \tag{9}$$

To make the integral of the probability density function on the truncated range equal to 1, the output probability density $f(x)$ of any real number $x \in [L, R]$ is multiplied by the normalization coefficient n_δ, which is calculated according to $n_\delta = 1/(1 - I_L - I_R)$, and a normalized probability density $p(x)$ is obtained, i.e.,

$$p(x | x \in [L, R]) = \frac{n_\delta}{2\sigma} e^{-|x-\delta|/\sigma}. \tag{10}$$

Correspondingly, the red/blue curve in Figure 2 represents the probability density function of the Laplace/truncated and normalized Laplace distribution, respectively. In the latter one, the output range of the perturbation result is considerably narrowed, and the probability of the output result near the true value remains the maximum. The truncated Laplace distribution curve is not symmetrical like the conventional Laplace distribution under certain parameter conditions, which leads to the deviation between the

expected value and the true degree. Nevertheless, because of the limitation of the output range, truncated Laplace has a smaller mean square error than the conventional mechanism under a low privacy budget. Thus, we can infer that the limitation of the length of output interval for user's degree perturbation will greatly affect the utility of the noisy degree vector and the accuracy of the community detection results.

Algorithm 1 describes the local perturbation process in detail. In particular, during each round of user community bipartition, user i will receive a bipartite grouping $\mathbf{G}$ of the user subset $\mathbf{U}$ including herself from the server for multiple times. Each time, she keeps her personal social data locally and calculates the corresponding true degree vector according to the grouping situation (Line 1). When delivering $\mathbf{G}$ to the users, the server will notify user i of the length of the truncated range, which is denoted as l. Based on this, user i generates two sequences of output intervals with the same length of truncation corresponding to $\mathbf{U}^1$ and $\mathbf{U}^2$ (Line 2) and obtains the output interval for each dimension according to her true degree vector (Line 3). Afterward, user i spends privacy budget ε to perturb each dimension. Through the true degree δ, the left boundary L and the right boundary R of the output interval, the scaling parameter σ, and the normalization coefficient n_δ, as well as the probability density function $p(x|x \in [L, R])$, can be calculated (Line 7). To ensure that the perturbation result is meaningful, the output probability of any integer in the output interval is calculated according to the data perturbation steps shown in lines 8–11 to make that any real result obtained by the truncated Laplace mechanism are rounded, which still satisfies ε-LDP for each query.

4.4. Refined EO Algorithm. As mentioned in Section 4.2.1, directly performing the grouping adjustment of the classical EO algorithm and the recalculation of user fitness in the application scenario of LDP will not only cause the excessive allocation of the privacy budget but also lead to a catastrophic increase in the cost of user-server communication. For solving this problem, we attempt to make full use of the uploaded data and propose to form an iterative process inside the server based on the perturbed degree vectors and the corresponding user grouping, thereby greatly reducing the total number of interactions between users and the server during the process of optimal bipartition for each community. Specifically, we assume that the degree vector of all users remains unchanged after any number of users are migrated to the other group; then, the change in fitness is only related to the calculable $a_{c(i)}$. After the move of user i, the sum of user degrees in the original group, where user i is located, needs to be subtracted by i's perturbed degree, and the sum of degrees in i's current group needs to be added with i's degree value accordingly. Therefore, the changes of $\widetilde{a}_1$ and $\widetilde{a}_2$ of the two groups can be calculated correspondingly, and the fitness of all users can also be updated according to (3).

In Algorithm 2, we describe the whole process consisting of the server-side migration of users in bipartite grouping and several interactions of degree query to finally obtain an optimal grouping with converged bipartite modularity.

Initially, after setting the length of the truncated range and the privacy budget of a single query, the server bisects the user subset $\mathbf{U}$ randomly into two groups (line 1), followed by sending all this information to the users. Each user substitutes the above privacy protection parameters into Algorithm 1 to obtain her perturbed degree vector and uploads it to the center (line 4). After that, the server aggregates these noisy data to directly calculate the total number of edges, $\widetilde{a}_1$ and $\widetilde{a}_2$, the fitness of each user, and the bipartition modularity of the initial grouping (lines 5–10).

Then, the server performs the grouping adjustment step shown in lines 12–21 of Algorithm 2. The user $m\prime$ with the lowest fitness is migrated to the other group. Based on the noisy degree vector uploaded by users and the fine-tuned $\widetilde{a}_1$ and $\widetilde{a}_2$ after the migration of $m\prime$ (lines 16–19), the fitness of all users is updated, and the user with the lowest fitness is found out and migrated again. This process will continue to iterate, during which the server only uses the perturbed degree vectors uploaded by users according to the initial random grouping and does not consume any additional privacy budget. From the classical EO algorithm, it can be inferred that the ultimate goal of the iteration is to make the fitness of all users in the final convergent grouping situation greater than 0, which means that the bipartition modularity no longer increases. Considering that this situation may not be reached in the end, the convergence conditions are relaxed. When the last two search results are the same user with the minimum fitness (line 14), the migration iteration of users can be stopped for this time.

Considering the degree vector of each user is in fact changing implicitly in the continuous adjustment of user grouping, the iterative migration of users based on the degree vectors of the initial grouping cannot derive the expected near-optimal bipartition of $\mathbf{U}$. For this problem, another iterative process of degree query between users and the server is constructed. After the server performs the steps in lines 12–21 based on the initial grouping and the corresponding degree vector, the formed convergent grouping will continue to be delivered to all users as the baseline grouping of a new query to obtain the next new perturbed degree vectors (line 22), and from it, the server will perform the user migration iterative process of the next time (in Algorithm 2, we take s to denote the number of degree queries). Therefore, a small iterative migration process is nested in a larger iterative process of degree query. In this way, the number of user interactions with the server and the consumption of the privacy budget can ultimately be greatly reduced. When the algorithm is executed until the bipartition modularity of user convergent grouping $\mathbf{G}$ does not increase (line 23), the ideal division result of the user subset $\mathbf{U}$ is obtained.

According to the description of the framework in Section 4.1, when each user subset $\mathbf{U}_{r-1}$ in the initial community division $\mathbf{C}_{r-1}$ of the rth round does not increase the total modularity after completing the optimal bipartition, the iteratively splitting process of user subset gets terminated.

Input: user i's adjacent bit vector $B_i \in \{0,1\}^N$, the division of the group $\mathbf{U}$ where user i is located $\mathbf{G} = \{\mathbf{U}^1, \mathbf{U}^2\}$, the truncated interval length given by the server l, privacy budget ε
Output: perturbed degree vector $\tilde{\boldsymbol{\delta}}_i$ of user i under the grouping $\mathbf{G}$
(1) Calculate the true degree vector $\boldsymbol{\delta}_i = \{\delta_i^1, \delta_i^2\}$ based on B_i and $\mathbf{G}$;
(2) Generate $\mathbf{I}_1$ and $\mathbf{I}_2$ corresponding to $\mathbf{U}^1$ and $\mathbf{U}^2$, respectively, according to l, which is $\mathbf{I}_1 = \{[0,l), [l,2l), \ldots, [\lfloor |\mathbf{U}^1|/l \rfloor \cdot l, |\mathbf{U}^1|\}$ and $\mathbf{I}_2 = \{[0,l), [l,2l), \ldots, [\lfloor |\mathbf{U}^2|/l \rfloor \cdot l, |\mathbf{U}^2|\}$;
(3) Find $[L_1, R_1)$ and $[L_2, R_2)$ in $\mathbf{I}_1$ and $\mathbf{I}_2$, respectively, to satisfy that $\delta_i^1 \in [L_1, R_1)$ and $\delta_i^2 \in [L_2, R_2)$;
(4) Set $\delta = \delta_i^1$ or δ_i^2, $L = L_1$ or L_2, $R = R_1$ or R_2, respectively;
(5) if $\delta = L$ and $\delta > 0$ then
(6) Randomly chose a truncated range between $[L, R]$ and $[L - l, L]$ (let $L = L - l$ $R = L$) with equal probability, then perturb δ within it;
(7) With ε, calculate σ and $p(x|x \in [L,R])$ according to equations (7)–(10);
(8) Calculate $\Pr(L) = \int_L^{L+0.5} p(x|x \in [L,R])\mathrm{d}x$, $\Pr(R) = \int_{R-0.5}^{R} p(x|x \in [L,R])\mathrm{d}x$;
(9) for $d \in [L+1, R-1]$ $(d \in \mathbb{N})$ do
(10) Calculate $\Pr(d) = \int_{d-0.5}^{d+0.5} p(x|x \in [L,R])\mathrm{d}x$;
(11) Sample a value according to the output probability of each integer in $[L, R]$ and assign it to the perturbed degree $\tilde{\delta}$;
(12) Obtain the final perturbed degree vector $\tilde{\boldsymbol{\delta}}_i$.

Algorithm 1: The implementation of truncated Laplace mechanism on the user side

Input: user subset $\mathbf{U}$, $B_i \in \{0,1\}^N$ $(\forall i \in \mathbf{U})$, truncated interval length l, single query privacy budget ε
Output: $\mathbf{U}$'s final bipartition result $\mathbf{G}_\mathrm{f} = \{\mathbf{U}_\mathbf{f}^1, \mathbf{U}_\mathbf{f}^2\}$, privacy cost $\varepsilon_{\mathrm{cost}}$
(1) Initialize random bisection of the server $\mathbf{G}_0 = \{\mathbf{U}_0^1, \mathbf{U}_0^2\}$, and set $s = 0$;
(2) The server delivers grouping $\mathbf{G}_s$ to users in $\mathbf{U}$;
(3) for $i \in \mathbf{U}$ do
(4) User i executes Algorithm 1 (with input parameters B_i, $\mathbf{G}_s$, l, and ε) and obtains $\tilde{\boldsymbol{\delta}}_i$ as well as $\tilde{\delta}_i = \tilde{\delta}_i^1 + \tilde{\delta}_i^2$ to upload to the server;
(5) The server calculates the total number of edges $\tilde{L} = \sum_i \tilde{\delta}_i/2$, as well as $\tilde{a}_1 = \sum_{j \in U_s^1} \tilde{\delta}_j/2\tilde{L}$ and $\tilde{a}_2 = \sum_{k \in \mathbf{U}_s^2} \tilde{\delta}_k/2\tilde{L}$;
(6) for $j \in \mathbf{U}_s^1$ do
(7) Calculate the fitness $\tilde{\lambda}_j = \tilde{\delta}_j^1/\tilde{\delta}_j - \tilde{a}_1$;
(8) for $k \in \mathbf{U}_s^2$ do
(9) Calculate the fitness $\tilde{\lambda}_k = \tilde{\delta}_k^2/\tilde{\delta}_k - \tilde{a}_2$;
(10) Calculate the bipartition modularity of $\mathbf{G}_s$, i.e., $\tilde{Q}_{\mathrm{b}(s)} = \sum_i \tilde{\delta}_i \tilde{\lambda}_i / 2\tilde{L}$;
(11) do
(12) Find the user with the lowest fitness value as $m\prime\,(\in \mathbf{U})$;
(13) Set $m = -1$, $\mathbf{U}^1 = \mathbf{U}_s^1$, $\mathbf{U}^2 = \mathbf{U}_s^2$;
(14) while $m \neq m'$ do
(15) $m = m'$;
(16) if $m \in \mathbf{U}^1$, then
(17) $\mathbf{U}^1 = \mathbf{U}^1 \backslash \{m\}$, $\tilde{a}_1 = \tilde{a}_1 - \tilde{\delta}_m/2\tilde{L}$; $\mathbf{U}^2 = \mathbf{U}^2 \cup \{m\}$, $\tilde{a}_2 = \tilde{a}_2 + \tilde{\delta}_m/2\tilde{L}$;
(18) else
(19) $\mathbf{U}^1 = \mathbf{U}^1 \cup \{m\}$, $\tilde{a}_1 = \tilde{a}_1 + \tilde{\delta}_m/2\tilde{L}$; $\mathbf{U}^2 = \mathbf{U}^2 \backslash \{m\}$, $\tilde{a}_2 = \tilde{a}_2 - \tilde{\delta}_m/2\tilde{L}$;
(20) Recalculate according to lines 6–9, and repeat the step in line 12;
(21) Get the stable grouping $\mathbf{G} = \{\mathbf{U}^1, \mathbf{U}^2\}$, let $s = s + 1$ and set $\mathbf{G}_s = \mathbf{G}$;
(22) Repeat the steps in lines 2–10;
(23) while $\tilde{Q}_{\mathrm{b}(s-1)} \leq \tilde{Q}_{\mathrm{b}(s)}$
(24) The final group $\mathbf{G}_\mathbf{f} = \mathbf{G}_s$, privacy cost $\varepsilon_{\mathrm{cost}} = s\varepsilon$.

Algorithm 2: User's group bipartition by refined EO.

Therefore, it is necessary to estimate the modularity gain $\Delta\tilde{Q}$ after each execution of Algorithm 2. To obtain unbiased estimation results, each user is required to consume additional privacy budget ε_f and utilize Laplace mechanism to add calibrated noise on her true degree vector, which is based on the optimal bipartition result of the user subset including her (also as $\mathbf{G}_\mathrm{f}$). According to this, the user degree in the subset $\mathbf{U}$ and the total number of edges $\tilde{L}$ of the user subset can be derived by referring to lines 4 and 5 in Algorithm 2. The total number of edges $\tilde{L}_1$ and $\tilde{L}_2$ within each one of the final groups can also be directly calculated, i.e.,

$$\tilde{L}_1 = \frac{1}{2} \sum_{j \in \mathbf{U}_\mathbf{f}^1} \tilde{\delta}_j^1, \tilde{L}_2 = \frac{1}{2} \sum_{k \in \mathbf{U}_\mathbf{f}^2} \tilde{\delta}_k^2. \tag{11}$$

We also use the noisy degree vectors with Laplace noise from the first round's optimal bipartition for all users to calculate the total degree of each user and the total number of edges in the entire network. Here, we slightly abuse the

notations by using $\widetilde{\delta}_i^{\mathrm{t}}$ and $\widetilde{L}_{\mathrm{t}}$ to denote them, respectively, and

$$\widetilde{L}_{\mathrm{t}} = \frac{1}{2}\left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\widetilde{\delta}_j^{\mathrm{t}} + \sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\widetilde{\delta}_k^{\mathrm{t}} + \sum_{i\in\mathbf{U}_0\setminus\mathbf{U}}\widetilde{\delta}_i^{\mathrm{t}}\right), \tag{12}$$

in which the three items on the right side are mutually independently perturbed with Laplace mechanism. According to the definition of modularity, the estimated gain of $\widetilde{Q}$ after the optimal bipartition of user subset $\mathbf{U}$ can be calculated, i.e.,

$$\begin{aligned}\Delta\widetilde{Q} &= \frac{\widetilde{L}_1}{\widetilde{L}_{\mathrm{t}}} - \left(\frac{\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\widetilde{\delta}_j^{\mathrm{t}}}{2\widetilde{L}_{\mathrm{t}}}\right)^2 + \frac{\widetilde{L}_2}{\widetilde{L}_{\mathrm{t}}} - \left(\frac{\sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\widetilde{\delta}_k^{\mathrm{t}}}{2\widetilde{L}_{\mathrm{t}}}\right)^2 - \left[\frac{\widetilde{L}}{\widetilde{L}_{\mathrm{t}}} - \left(\frac{\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\widetilde{\delta}_j^{\mathrm{t}} + \sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\widetilde{\delta}_k^{\mathrm{t}}}{2\widetilde{L}_{\mathrm{t}}}\right)^2\right], \\ &= \frac{\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\widetilde{\delta}_j^{\mathrm{t}}\sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\widetilde{\delta}_k^{\mathrm{t}} - \widetilde{L}_{\mathrm{t}}\left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\widetilde{\delta}_j^2 + \sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\widetilde{\delta}_k^1\right)}{2\left(\widetilde{L}_{\mathrm{t}}\right)^2}.\end{aligned} \tag{13}$$

Since each item of equation (13) is an unbiased estimation of its corresponding true value, it can be observed that if its numerator is larger than 0 by the value of its standard deviation, the optimal bipartition will cause positive gain in the total modularity with an adequate probability. Thus, considering that the Laplace noise of $\widetilde{\delta}_i^{\mathrm{t}}$ and $\widetilde{L}_{\mathrm{t}}$ is independent of that injected to $\widetilde{\delta}_j^2$ and $\widetilde{\delta}_k^1$ and by using equation (12), we attempt to derive the specific form of its variance as follows:

$$\begin{aligned}\mathrm{Var}\left[\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\widetilde{\delta}_j^{\mathrm{t}}\sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\widetilde{\delta}_k^{\mathrm{t}} - \widetilde{L}_{\mathrm{t}}\left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\widetilde{\delta}_j^2 + \sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\widetilde{\delta}_k^1\right)\right] &= E\left[\left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\widetilde{\delta}_j^{\mathrm{t}}\right)^2\right]E\left[\left(\sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\widetilde{\delta}_k^{\mathrm{t}}\right)^2\right] - E\left[2\widetilde{L}_{\mathrm{t}}\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\widetilde{\delta}_j^{\mathrm{t}}\sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\widetilde{\delta}_k^{\mathrm{t}}\right]\left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\delta_j^2 + \sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\delta_k^1\right) \\ &\quad + E\left[\left(\widetilde{L}_{\mathrm{t}}\right)^2\right]E\left[\left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\widetilde{\delta}_j^2 + \sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\widetilde{\delta}_k^1\right)^2\right] - \left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\delta_j^{\mathrm{t}}\sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\delta_k^{\mathrm{t}} - L_{\mathrm{t}}\left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\delta_j^2 + \sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\delta_k^1\right)\right)^2 \\ &= \mathrm{Var}\left[\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\widetilde{\delta}_j^{\mathrm{t}}\right]\left(\sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\delta_k^{\mathrm{t}}\right)^2 + \mathrm{Var}\left[\sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\widetilde{\delta}_k^{\mathrm{t}}\right]\left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\delta_j^{\mathrm{t}}\right)^2 + \mathrm{Var}\left[\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\widetilde{\delta}_j^{\mathrm{t}}\right]\mathrm{Var}\left[\sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\widetilde{\delta}_k^{\mathrm{t}}\right] \\ &\quad - \left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\delta_j^2 + \sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\delta_k^1\right)\left[\mathrm{Var}\left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\widetilde{\delta}_j^{\mathrm{t}}\right)\sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\delta_k^{\mathrm{t}} + \mathrm{Var}\left(\sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\widetilde{\delta}_k^{\mathrm{t}}\right)\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\delta_j^{\mathrm{t}}\right] + \mathrm{Var}\left(\widetilde{L}_{\mathrm{t}}\right) \\ &\quad \cdot\left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\delta_j^2 + \sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\delta_k^1\right)^2 + \left(L_{\mathrm{t}}\right)^2\mathrm{Var}\left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\widetilde{\delta}_j^2 + \sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\widetilde{\delta}_k^1\right) + \mathrm{Var}\left(\widetilde{L}_{\mathrm{t}}\right)\mathrm{Var}\left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\widetilde{\delta}_j^2 + \sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\widetilde{\delta}_k^1\right).\end{aligned} \tag{14}$$

Noticing that the variance of Laplace noise with privacy budget ε and sensitivity Δf is $2(\Delta f/\varepsilon)^2$, we can solve all the variance item in equation (14) and reach the final expression as

$$\begin{aligned}\mathrm{Var}\left[\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\widetilde{\delta}_j^{\mathrm{t}}\sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\widetilde{\delta}_k^{\mathrm{t}} - \widetilde{L}_{\mathrm{t}}\left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\widetilde{\delta}_j^2 + \sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\widetilde{\delta}_k^1\right)\right] &= \frac{4\left|\mathbf{U}_{\mathrm{f}}^1\right|}{\varepsilon_{\mathrm{f}}^2}\left(\sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\delta_k^{\mathrm{t}}\right)^2 + \frac{4\left|\mathbf{U}_{\mathrm{f}}^2\right|}{\varepsilon_{\mathrm{f}}^2}\left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\delta_j^{\mathrm{t}}\right)^2 + \frac{16\left|\mathbf{U}_{\mathrm{f}}^1\right|\left|\mathbf{U}_{\mathrm{f}}^2\right|}{\varepsilon_{\mathrm{f}}^4} - \left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\delta_j^2 + \sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\delta_k^1\right) \\ &\quad \cdot\left[\frac{4\left|\mathbf{U}_{\mathrm{f}}^1\right|}{\varepsilon_{\mathrm{f}}^2}\sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\delta_k^{\mathrm{t}} + \frac{4\left|\mathbf{U}_{\mathrm{f}}^2\right|}{\varepsilon_{\mathrm{f}}^2}\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\delta_j^{\mathrm{t}}\right] + \frac{N}{\varepsilon_{\mathrm{f}}^2}\left(\sum_{j\in\mathbf{U}_{\mathrm{f}}^1}\delta_j^2 + \sum_{k\in\mathbf{U}_{\mathrm{f}}^2}\delta_k^1\right)^2 + \frac{2|\mathbf{U}|\left(L_{\mathrm{t}}\right)^2}{\varepsilon_{\mathrm{f}}^2} + \frac{2N|\mathbf{U}|}{\varepsilon_{\mathrm{f}}^4}.\end{aligned} \tag{15}$$

After the server obtains the optimal bipartition of **U** and receives corresponding noisy degree vector from each user in **U**, also with their noisy total degrees as well as the total edges, we can simply calculate the estimation value of equation (13) numerator. If it is greater than the square root of the right side of equation (15), the modularity gain caused by **U**'s division is considered positive, and the bipartition $\{\mathbf{U}_i^1, \mathbf{U}_i^2\}$ will be accepted to replace $\{\mathbf{U}\}$ in the community division of the entire graph.

4.5. Theoretical Analysis

4.5.1. Proof of ε-LDP Guarantee. This section will prove that the truncated Laplace mechanism proposed in Section 4.3 satisfies ε-LDP.

In the process of degree perturbation by users, it is assumed that the local relationship data of each user in two adjacent graphs (differed by any single edge) are D_1 and D_2, which are either the same or different by one bit. The function $f(D)$ is the degree query function. Therefore, the local sensitivity is $\Delta f = \max_{D_1,D_2} |f(D_1) - f(D_2)| = 1$. According to the step of truncated interval selection, it is assumed that both $f(D_1)$ and $f(D_2)$ are in the interval $[L, R]$; thus, any result $s \in \mathbb{R}$ obtained by the perturbation also satisfies $s \in [L, R]$. The truncated Laplace mechanism is set to be M. According to Definition 1, it is necessary to prove that the following inequality is always valid:

$$e^{-\varepsilon} \le \frac{\Pr[M(f(D_1)) = s]}{\Pr[M(f(D_2)) = s]} \le e^{\varepsilon}. \tag{16}$$

Without loss of generality, as shown in Figure 3, it is assumed that $L \le f(D_2) \le f(D_1) \le R$, and let $\Delta L_1 = |f(D_1) - R|$, $\Delta R_1 = |f(D_1) - L|$, $\Delta L_2 = |f(D_2) - R|$, $\Delta R_2 = |f(D_2) - L|$, and $|f(D_1) - f(D_2)| = i\Delta f\ (i = 0, 1)$. The following theorems are given and proved below.

Theorem 1. *With the given truncated interval $[L, R]$ and the privacy budget ε, as well as the user's local degree $f(D_1)$ or $f(D_2)$, if the scaling parameter σ in the Laplace distribution satisfies $\sigma = 2\Delta f/\varepsilon$, the results obtained by perturbing the true degrees $f(D_1)$ or $f(D_2)$ according to the probability distribution of equation (10) must satisfy ε-LDP.*

Proof. Replace the probability expression in equation (16) with equation (10). Then, replace I_{L2} and I_{R2} with I_{L1} and I_{R1}, and use the triangle inequality, and we can obtain

$$\frac{n_{\delta 1}/2\sigma \cdot e^{-|s-f(D_1)|/\sigma}}{n_{\delta 2}/2\sigma \cdot e^{-|s-f(D_2)|/\sigma}} \le \frac{1 - \left(I_{L1}e^{i\Delta f/\sigma} + I_{R1}e^{-i\Delta f/\sigma}\right)}{1 - I_{L1} - I_{R1}} e^{i\Delta f/\sigma}. \tag{17}$$

To prove that the left side is smaller than e^{ε}, it is attempted to prove that the right side does not exceed $e^{i\varepsilon}$ on $i \in [0, 1]$. Thus, taking $i \in [0, 1]$ as the independent variable, let

$$\begin{aligned} F(i) &= \frac{1 - \left(I_{L1}e^{i\Delta f/\sigma} + I_{R1}e^{-i\Delta f/\sigma}\right)}{1 - I_{L1} - I_{R1}}, \\ R(i) &= e^{i(\varepsilon - \Delta f/\sigma)}. \end{aligned} \tag{18}$$

By careful observation, when $i = 0$, $F(i) = R(i) = 1$. If let $F(i) \le R(i)$, when $i \ge 0$, we should require that $F'(0) \le R'(0)$. By solving the inequality, the value range of σ is obtained, i.e.,

$$\sigma \ge \frac{\Delta f}{\varepsilon} \frac{2I_{L1} - 1}{I_{L1} + I_{R1} - 1}. \tag{19}$$

In order to ensure that $F'(i)$ does not exceed $R'(i)$ when $i > 0$, the secondary derivatives of these two functions are calculated as follows:

$$\begin{aligned} F''(i) &= -\left(\frac{\Delta f}{\sigma}\right)^2 \frac{I_{R1}e^{-i\Delta f/\sigma} + I_{L1}e^{i\Delta f/\sigma}}{1 - I_{L1} - I_{R1}}, \\ R''(i) &= \left(\varepsilon - \frac{\Delta f}{\sigma}\right)^2 e^{i(\varepsilon - \Delta f/\sigma)}. \end{aligned} \tag{20}$$

According to equation (8) and equation (9), it can be known that $I_{L1} + I_{R1} < 1$ is always valid. Therefore, when $i > 0$, we can infer that $F''(i) < 0$ and $R''(i) \ge 0$, which means that $F'(i)$ is monotonically decreasing and $R'(i)$ is monotonically increasing. Because σ satisfies equation (19) to ensure that $F'(0) \le R'(0)$, $F'(i) \le R'(i)$ is always true for $i \ge 0$. Thus, it can be inferred that $F(i) \le R(i)$ always holds on $i \in [0, 1]$, and accordingly,

$$\frac{\Pr[M(f(D_1)) = s]}{\Pr[M(f(D_2)) = s]} = \frac{n_{\delta 1}/2\sigma \cdot e^{-|s-f(D_1)|/\sigma}}{n_{\delta 2}/2\sigma \cdot e^{-|s-f(D_2)|/\sigma}} \le e^{i\varepsilon} \le e^{\varepsilon}. \tag{21}$$

Similarly, the above deducing process can also be used to prove the validity of the left half of equation (16) under the following condition:

$$\sigma \ge \frac{\Delta f}{\varepsilon} \frac{2I_{R2} - 1}{I_{L2} + I_{R2} - 1}. \tag{22}$$

Therefore, if truncated Laplace mechanism is to strictly satisfy ε-LDP, σ must simultaneously satisfy

$$\begin{aligned} \sigma &\ge \frac{\Delta f}{\varepsilon} \frac{2I_{L1} - 1}{I_{L1} + I_{R1} - 1} = \frac{2\Delta f}{\varepsilon} \frac{1}{1 + \left(1 - e^{-\Delta L_1/\sigma}\right)/\left(1 - e^{-\Delta R_1/\sigma}\right)}, \\ \sigma &\ge \frac{\Delta f}{\varepsilon} \frac{2I_{R2} - 1}{I_{L2} + I_{R2} - 1} = \frac{2\Delta f}{\varepsilon} \frac{1}{1 + \left(1 - e^{-\Delta R_2/\sigma}\right)/\left(1 - e^{-\Delta L_2/\sigma}\right)}. \end{aligned} \tag{23}$$

Finally, extreme cases are used to find the strict lower bound of σ, which happens when $\Delta L_1 = 0$ or $\Delta R_2 = 0$, and accordingly,

$$\sigma \ge \frac{2\Delta f}{\varepsilon}. \tag{24}$$

Therefore, if σ satisfies equation (24), it must also satisfy equation (23). The conclusion is thus proved. □

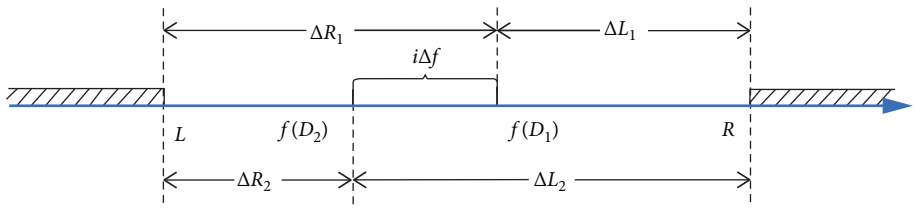


Figure 3: The diagram of the truncated range for degree perturbation.

5. Experiments

In this section, the experiments are performed to evaluate our proposed method.

5.1. Experimental Methods

5.1.1. Datasets. Two datasets published in the Stanford Network Analysis Project (SNAP) are used to conduct the experiments:

(1) Facebook dataset [37]: this dataset contains 4039 Facebook users and 88234 relationship edges (undirected edges) formed among them. In addition, this dataset is one of the classic datasets employed for complex network community detection.

(2) Facebook page network dataset about the government [38]: this dataset involves 7057 web pages about government information, and the edges between the web pages represent their mutual likes (undirected edges). After removing the self-loop edges in the original data, the total number of edges in the network is 89429.

5.1.2. Compared Methods. In order to evaluate the performance of LDPCD, the proposed method is compared with the following two methods.

(1) LDPGen [19]: with this method, a synthetic private social graph under LDP protection is generated. Besides, the experiment in our paper still follows the parameter settings of LDPGen, in which two times of queries on user degree vectors and user clustering based on the k-means method in terms of degree vectors are implemented, and the total privacy budget is evenly distributed. Then, based on the generated synthetic social graph, the Louvain community detection algorithm is finally adopted.

(2) LFGDPR [20]: using this method, a variety of statistical graph metrics, including clustering coefficient distribution and subgraph modularity, are estimated under LDP. In accordance with all its technical details, the experiment in our paper optimally allocates the total privacy budget and then perturbs the user's total degree and adjacency bit vector, respectively. Furthermore, according to the proposed Louvain community detection algorithm based on LFGDPR, the network community is divided.

5.1.3. Utility Metrics. According to Section 3.4, the specific meanings of the three utility metrics, namely, modularity, ARI (adjusted random index), and AMI (adjusted mutual information), are elaborated.

Modularity: the function of $\mathrm{modu}(\mathbf{C})$ is to calculate the total modularity of a community division result $\mathbf{C}$ based on the original real network. Therefore, $\mathrm{modu}(\mathbf{C}_p)$ is taken as the evaluation index for the quality of community detection results under privacy protection, with the value range of $[-1/2, 1]$.

ARI and AMI: given all users $\{u_1, \ldots, u_N\}$ and their two grouping conditions $X = \{x_1, \ldots, x_u\}$ and $Y = \{y_1, \ldots, y_v\}$, n_{ij} represents the number of users shared by group x_i and group y_j, that is, $n_{ij} = |X_i \cap Y_j|$, and $1 \le i \le u$ and $1 \le j \le v$. In addition, let $a_i = \sum_j n_{ij}$ and $b_j = \sum_i n_{ij}$, and define the index to measure the similarity between groupings X and Y as

$$\mathrm{ARI}(X,Y) = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{N}{2}}{1/2\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{N}{2}}, \tag{25}$$

$$\mathrm{AMI}(X,Y) = \sum_{i=1}^{u}\sum_{j=1}^{v}\sum_{n_{ij}=\left(a_i+b_j-N\right)^+}^{\min\left(a_i,b_j\right)} \frac{n_{ij}}{N}\log\left(\frac{N\cdot n_{ij}}{a_i b_j}\right) \times \frac{a_i!b_j!\left(N-a_i\right)!\left(N-b_j\right)!}{N!n_{ij}!\left(a_i-n_{ij}\right)!\left(b_j-n_{ij}\right)!\left(N-a_i-b_j+n_{ij}\right)!}.$$

Among them, $(a_i + b_j - N)^+$ means $\max(1, a_i + b_j - N)$. Specifically, ARI represents the frequency of agreements between the two obtained groupings over all element pairs, and AMI quantitatively refers to the amount of information shared by the two groupings X and Y. In the case of the same grouping, AMI is usually higher than ARI, both with the value range of $[0, 1]$. Higher values of ARI and AMI mean that two groupings of the same user set are more similar. Therefore, in the experiment, higher $\mathrm{ARI}(\mathbf{C}_p, \mathbf{C}_t)$ or $\mathrm{AMI}(\mathbf{C}_p, \mathbf{C}_t)$ indicates that the community detection result under LDP has higher accuracy.

5.1.4. Parameter Settings. In this paper, the parameter settings of the LDP-based community detection algorithm are as follows.

As shown in Algorithm 2, each time the server queries user degree, the assigned privacy budget is equal, which is denoted as ε. Other than that, calculating the total modularity gain also requires to be allocated a certain privacy budget ε_f so as to query the degree vector perturbed by Laplace mechanism under the final bipartition $\mathbf{G}_f$. Thus, we set the value range of ε_f within 0.02 to 0.1, while in contrast, that of ε is from 0 to 0.05. If the total number of rounds for the division of user communities is r and the number of times that user i uploads data to the server in each round of bipartite division is listed as $\{s_1^i, s_2^i, \ldots, s_r^i\}$, the total privacy cost is $\varepsilon_{\text{total}} = \max_{j \in [0, N-1]} (\varepsilon \sum_{n=1}^{r} s_n^j + r\varepsilon_f)$.

Besides, in the truncated Laplace mechanism, the length of truncated range l is one of the key parameters that needs to be explored. On the one hand, the degree distribution of complex social networks is usually expressed as power-law distribution [29], in which the total degree of most users is below the average. Therefore, in this experiment, the maximum value of l for the Facebook dataset (average degree of 42) and the Government dataset (average degree of 25) are set to 30 and 20, respectively. On the other hand, excessively small l will cause the perturbed degree to be too close to the true one, which can lead to the privacy disclosure. Considering that, the minimum value of l is set to 5.

In terms of the total privacy budget/cost, considering when the value of it is greater than 2.5 in [20], the probability of each bit in the adjacency vector not to be flipped exceeds 90%, which will expose the privacy concerning most of the user's connected edges (for the Facebook dataset, the allocated privacy budget for perturbing user's bit vector is 2.225. Since the ratio of bit flipping probability to the unflipping probability is $e^{-2.225}$ according to the definition of LDP, we can easily understand that the perturbed bit remains unchanged with a probability of $e^{2.225}/(1 + e^{2.225}) = 0.902472$). In this case, in order to ensure an appropriate protection strength of LDP, the total privacy budget in the contrast experiments is limited to 0.1–2.5.

5.1.5. Experimental Setting. Based on the given total privacy budget $\varepsilon_{\text{total}}$, LDPGen and LFGDPR are separately tested 100 times in our experiment. Then, the average values of modularity, ARI and AMI, from all community detection results are taken as the final result. While for LDPCD, the average result of the three measurement metrics obtained by 100 times of experiments with their actual total privacy cost in the range of $\varepsilon_{\text{total}} \pm 0.05$ is considered as the final result of our method.

5.2. Experimental Results. In this part of the paper, we will present detailed experimental results and corresponding analysis to demonstrate the feasibility of our proposed method. Firstly, some general but necessary data are enumerated, which give a brief outline of the comparison between the classical EO algorithm and LDPCD when they are separately employed in the datasets. Then, deeper discussions about the influence of parameters and the contrast result of LDPCD with two state-of-the-art methods will be elaborated according to the figures.

For the Facebook (Government) dataset, although the average modularity of the community partition results by classical EO algorithm reaches up to 0.813 (0.682) (see Figure 4(b)), the total times of user migration and recalculation of degree vectors exceed 20000 (35000). In contrast, under different parameter settings of LDPCD, the average modularity is within the range of 0.51–0.79 (0.31–0.63), with the total times of degree queries from 25 to 50 (from 40 to 70). In terms of the community numbers, the EO algorithm finally outputs 15 (29) user subsets as the ground truth, while LDPCD partitions all users into 8–16 (15–32) groups. From the abovementioned data, it can be obviously observed that our method simplifies the implementation of EO community divisive algorithm to a considerable extent and significantly reduce the communication cost of user-server interactions, which, in the same time, obtains utility-guaranteed results similar to the ground truth.

5.2.1. Questions about the Experiment. In the following sections, the effectiveness of LDPCD is verified based on experimental results. Besides, through the results, the following two questions are answered:

(1) EQ **1**: how do the length of truncated range and the total privacy cost affect the accuracy of community detection of LDPCD separately?

(2) EQ **2**: what are the advantages of the proposed method LDPCD compared with the existing methods under privacy protection of the same strength?

5.2.2. Influence of Experimental Parameters on the Results. From Figure 4, it can be seen that the length of the truncated output range has a more significant impact on the community division result than the total privacy cost. With the same privacy cost in the Facebook dataset (Government dataset), the average value concerning maximal changes of modularity, ARI, and AMI under the influence of l is 0.242, 0.322, and 0.367 (0.272, 0.173, and 0.253); while under the same output interval length, the average value of maximal changes in modularity, ARI, and AMI under the influence of total privacy cost $\varepsilon_{\text{total}}$ is

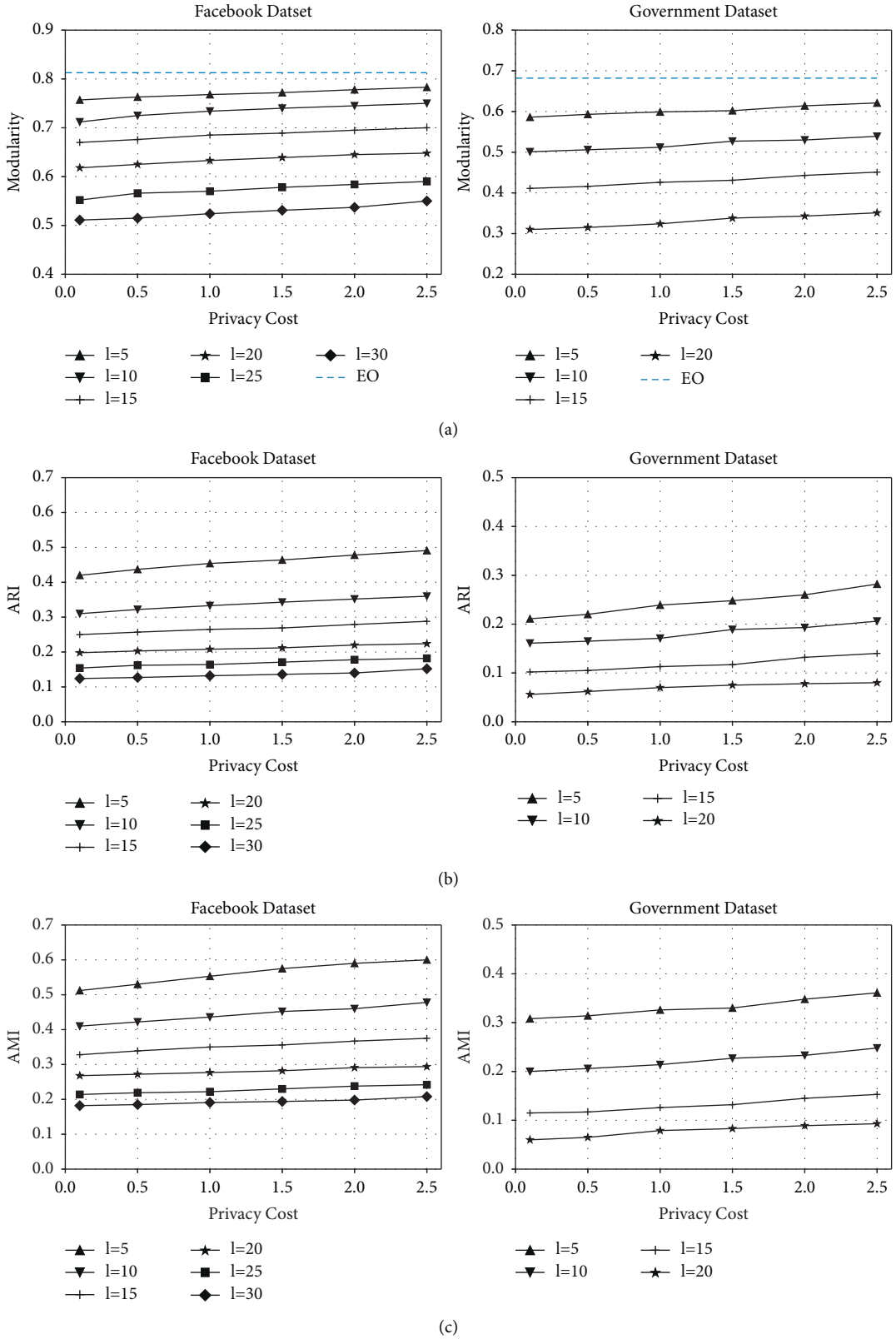


Figure 4: The experimental results of LDPCD on the Facebook and Government datasets (the bold dashed lines in Figure 4(a) indicate that the modularity result based on nonprivate EO algorithm in the Facebook/Government dataset is 0.813/0.682, respectively). (a) Modularity. (b) ARI. (c) AMI.

0.034, 0.040, and 0.047 (0.039, 0.045, and 0.043), as shown in Figures 4(a), 4(b), and 4(c), respectively. For these observed results, there are two main reasons. Firstly, with the introduction of the truncated Laplace mechanism, the variance of degree and the expected error of fitness $\widetilde{\lambda}$ are limited and closely related to l. Besides, the smaller the l, the smaller the perturbation noise of the degree vector and the total degree. If the true degree is large enough when compared with l, the deviation of fitness $\widetilde{\lambda}$ relative to its true value will be significantly reduced, and the probability of incorrect migration of the user during grouping adjustment will also decrease. As a result, the utility of community division will improve to a great extent with the decrease of l. Secondly, since multiple rounds of bipartition and several times of degree query in each round will consume privacy budget; even if the final total privacy cost experiences obvious increase, the privacy budget allocated for a single query remains relatively limited. Thus, the accuracy of the perturbation result by truncated Laplace mechanism is slightly enhanced (as shown in Table 2). In this case, the accuracy of community detection increases steadily with the rise of total privacy cost.

Moreover, by comparing the graphs in Figures 4(a), 4(b), and 4(c) separately, it can be seen that, under the same l and $\varepsilon_{\text{total}}$, the community detection results of the Facebook dataset are more similar to their ground truth than the Government dataset. The main reason is that the average degree of users in the Facebook dataset is higher. With similar total number of edges, the user nodes in the Government dataset is 1.75 times that of the Facebook dataset and its network structure is sparser; thus, the nodes with small total degree account for a larger proportion in it (for example, the ratio of nodes with their degree below 5 and 10 in the Facebook/Government dataset are 0.113 and 0.238/0.232 and 0.398, respectively). Therefore, under the same setting of l, more nodes in the Government dataset will be distributed in the output interval very close to $[0, l]$. Meanwhile, we note that the closer the output interval to $[0, l]$ is, the greater the deviation of the noisy fitness $\widetilde{\lambda}$ from its true value will be, resulting in a larger probability of incorrect node migration during the grouping adjustment. In that case, under the same privacy parameter settings, the utility of the community division result of the Government dataset will decrease more than that of the Facebook dataset.

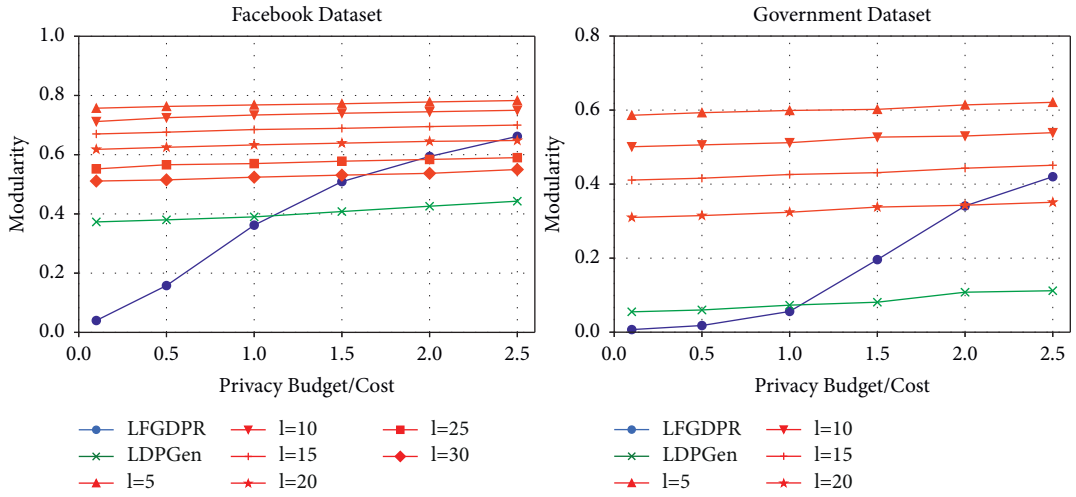
5.2.3. Comparison with the State-of-the-Art Methods. In this section, the experimental results of our proposed method are compared with those of the classical methods, namely, LDPGen [19] and LFGDPR [20]. The contrast results are shown in Figure 5.

Through a comprehensive analysis in Figure 5, it can be found that the result of LDPCD under all settings of l is better than that of the LDPGen method under the same total privacy budget/cost. The similarity between LDPGen and our method is that both enquire the user degree vector in a certain grouping situation, and the processed relationship data are all coarse-grained statistics, thus inevitably leading to the loss of some local information in the original social graph. Whereas, the difference is that LDPGen clusters users from the perspective of the similarity of the degree vector instead of their contribution to the total modularity, and based on the final grouping, a synthetic social graph is generated by the Chung-Lu probability model for further analysis of community structure. However, the user clustering and graph generation methods result in low utility of the community division under LDP, for the Chung-Lu model randomly connects user node pairs in the same group or different groups, which weakens the distinctiveness of densely connected node clusters in the original network and destroys the community features. This can be observed in Figures 5(b) and 5(c) that with the increase of the total privacy budget, even if the accuracy of the degree vector is gradually improved, the utility of the community detection result on the synthetic social graph is still in a low state and not significantly enhanced. In this case, it indicates that LDPGen cannot well balance the relationship between the strength of privacy protection and the accuracy of community mining results. In contrast, by using LDPCD, the user's contribution to the total modularity is calculated with the degree vector, and the community structure of the original network is gradually restored through multiple times of degree query and grouping adjustment. Otherwise, as mentioned in Section 5.2.2 that l plays a leading role in the experimental results, the utility of the community division obtained by LDPCD can maintain a relatively high level under different total privacy costs.

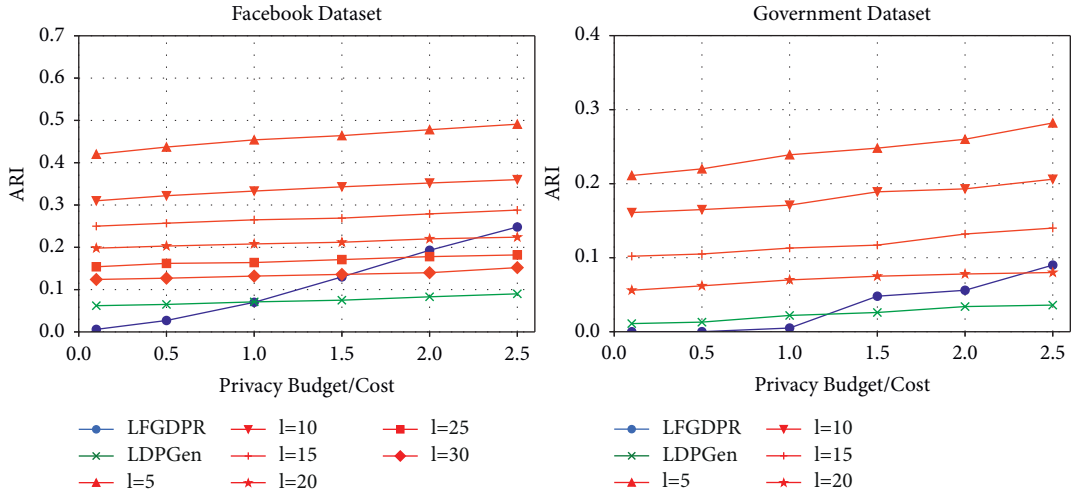
For LFGDPR, the perturbation object is the user's total degree and the most fine-grained adjacent bit. Since the processed data involve sensitive neighboring information, the privacy budget in the LFGDPR method should be adjusted to a small value to satisfy the requirement of sufficient privacy protection strength. As shown in Figures 5(a), 5(b), and 5(c), when the total privacy budget/cost of the Facebook dataset and the Government dataset is below 1.5 and 2.0, respectively, the accuracy of the community detection results given by LDPCD is higher than that of the LFGDPR method under all settings of l. Especially, when $\varepsilon_{\text{total}}$ is less than 1.0 and the privacy protection is strong enough, LDPCD has obvious advantages over LFGDPR. When $\varepsilon_{\text{total}} = 1$, the worst ARI/AMI value of the community detection result of the Facebook (Government) dataset of our proposed method is 1.89/2.03 (14.00/11.97) times that of LFGDPR (see Figures 5(b) and 5(c)), which also reflects that the LFGDPR method with low privacy budget has unsatisfactory effect on community detection in sparser social networks. When the privacy budget/cost is higher ($\varepsilon_{\text{total}} > 2.0$), the utility of the community detection result by LFGDPR is slightly better than that of some groups of experiments with larger l. However, we should note that when LFGDPR is adopted in the Facebook dataset (the Government dataset) with $\varepsilon_{\text{total}} = 2.0$, the probability of a single bit remaining unflipped is 5.81 (5.26) times than that of being flipped, respectively. This means LFGDPR is more liable to expose privacy under such privacy parameter settings. Thus, in contrast, LDPCD is superior to the LFGDPR method in terms of both the utility of the community detection results and the strength of privacy protection.

Table 2: The standard deviation of the noise of the truncated Laplace mechanism under different parameter settings of the Facebook dataset.

-		$l = 5$	$l = 10$	$l = 15$	$l = 20$	$l = 25$	$l = 30$
$\varepsilon_{total} = 0.1$	$\varepsilon = 0.0002$	2.166	4.160	6.152	8.144	10.136	12.127
$\varepsilon_{total} = 0.5$	$\varepsilon = 0.007$	2.161	4.144	6.117	8.082	10.038	11.987
$\varepsilon_{total} = 1.0$	$\varepsilon = 0.016$	2.156	4.122	6.070	7.998	9.909	11.802
$\varepsilon_{total} = 1.5$	$\varepsilon = 0.02$	2.153	4.113	6.049	7.962	9.852	11.719
$\varepsilon_{total} = 2.0$	$\varepsilon = 0.03$	2.147	4.082	5.996	7.869	9.708	11.514
$\varepsilon_{total} = 2.5$	$\varepsilon = 0.04$	2.141	4.065	5.944	7.777	9.565	11.308

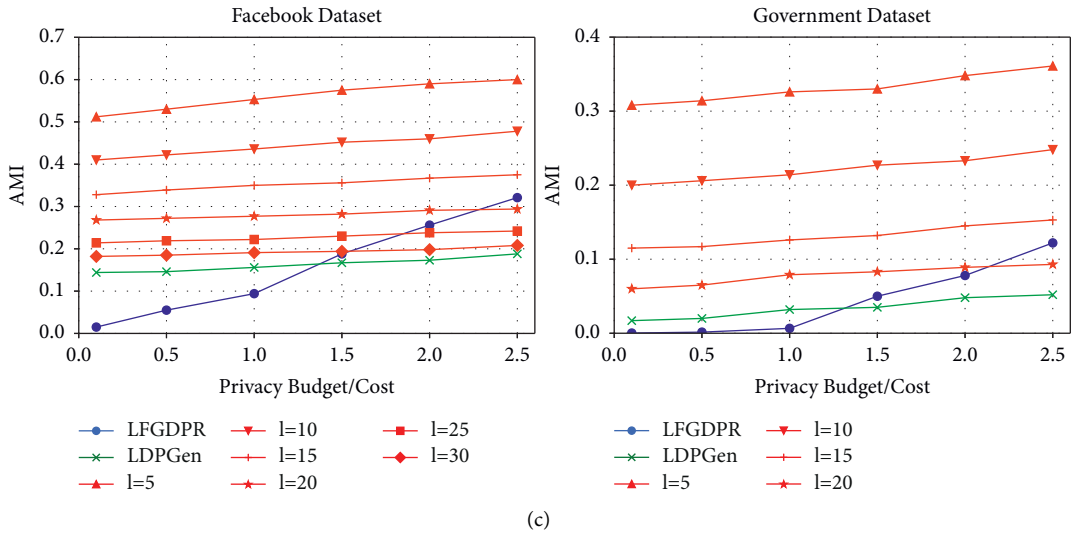


(a)



(b)

Figure 5: Continued.



(c)

Figure 5: The results obtained from the comparative experiment of LDPGen, LFGDPR, and LDPCD. (a) Modularity. (b) ARI. (c) AMI.

Furthermore, it should be emphasized that although the community detection accuracy of LDPCD improves slightly with the increase of the total privacy cost under a fixed truncated interval length l, in practical applications, appropriately increasing l can be considered when $\varepsilon_{\text{total}}$ is low while reducing l can be taken into account when $\varepsilon_{\text{total}}$ is high, so as to achieve a better tradeoff between the strength of privacy protection and the quality of community division results.

6. Conclusion

In this paper, LDPCD, a novel community detection method, is proposed based on the local differential privacy model. In the framework of LDPCD, the truncated Laplace mechanism with local differential privacy is employed to enhance the accuracy of user perturbation data. Other than that, by refining the community divisive algorithm based on extremal optimization, the number of interactions between users and the server is reduced, thus reducing the total privacy cost and ensuring strong privacy protection. Based on the above data perturbation and community detection algorithms, the community detection results with high utility are finally obtained. Furthermore, according to the experimental results on two real-world datasets, it can be concluded that LDPCD has the same or higher accuracy of community detection compared with the state-of-the-art methods under different settings of privacy protection parameters. In addition, LDPCD is featured with obvious superiority under strong privacy protection.

Data Availability

The experimental data used to support the findings of this study are available upon request to the author.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding this work.

References

[1] X. Cheng, S. Su, S. Xu, L. Xiong, K. Xiao, and M. Zhao, "A two-phase algorithm for differentially private frequent subgraph mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 8, pp. 1411–1425, 2018.

[2] M. Kuramochi and G. Karypis, "Frequent subgraph discovery," in *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM)*, pp. 313–320, San Jose, USA, November 2001.

[3] F. Chen, Z. Chen, X. Wang, and Z. Yuan, "The average path length of scale free networks," *Communications in Nonlinear Science and Numerical Simulation*, vol. 13, no. 7, pp. 1405–1410, 2008.

[4] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[5] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Physical review E, Statistical, nonlinear, and soft matter physics*, vol. 72, no. 2, Article ID 027104, 2005.

[6] J. Yang, J. J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in *Proceedings of the 2013 IEEE 13th International Conference on Data Mining (ICDM)*, pp. 1151–1156, Dallas, USA, December 2013.

[7] Z. Jorgensen and T. Yu, "A privacy-preserving framework for personalized, social recommendations," in *Proceedings of the 17th International Conference on Extending Database Technology (EDBT)*, pp. 571–582, Athens, Greece, March 2014.

[8] C. Wei, S. Ji, C. Liu, W. Chen, and T. Wang, "AsgLDP: collecting and generating decentralized attributed graphs with local differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3239–3254, 2020.

[9] P. Liu, Y. Xu, Q. Jiang et al., "Local differential privacy for social network publishing," *Neurocomputing*, vol. 391, pp. 273–279, 2020.

[10] B. Stephanie, Facebook Scandal a 'Game Changer' in Data Privacy Regulation, Bloomberg, April 2018.

[11] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proceedings of the 54th*

Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 429–438, Berkeley, USA, October 2013.

[12] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 1054–1067, Scottsdale, AZ, USA, November 2014.

[13] T. T. Nguyên, X. Xiao, and Y. Yang, "Collecting and analyzing data from smart device users with local differential privacy," 2016, https://arxiv.org/abs/1606.05053.

[14] X. Ren, C.-M. Yu, W. Yu et al., "Lopub high-dimensional crowdsourced data publication with local differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2151–2166, 2018.

[15] U. Stemmer, "Locally private k-means clustering," in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 548–559, Salt Lake City, USA, January 2020.

[16] D. Wang and J. Xu, "On sparse linear regression in the local differential privacy model," *IEEE Transactions on Information Theory*, vol. 67, no. 2, pp. 1182–1200, 2021.

[17] T. Wang, J. Blocki, N. Li, and S. Jha, "Locally differentially private protocols for frequency estimation," in *Proceedings of the 26th USENIX Security Symposium*, pp. 729–745, Vancouver, Canada, August 2017.

[18] T. Wang, N. Li, and S. Jha, "Locally Differentially Private Frequent Itemset Mining," in *Proceedings of the 2018 IEEE Symposium on Security and Privacy*, pp. 127–143, San Francisco, USA, May 2018.

[19] Z. Qin, T. Yu, and Y. Yang, "Generating synthetic decentralized social graphs with local differential privacy," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 425–438, Dallas, USA, October 2017.

[20] Q. Ye, H. Hu, M. H. Au, X. Meng, and X. Xiao, "LF-GDPR: A framework for estimating graph metrics with local differential privacy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 8, p. 1, 2020.

[21] M. Yang, L. Lyu, and J. Zhao, "Local differential privacy and its applications: a comprehensive survey," 2020, https://arxiv.org/abs/2008.03686.

[22] T. Gao, F. Li, Y. Chen, and X. Zou, "Local differential privately anonymizing online social networks under HRG-based model," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 4, pp. 1009–1020, 2018.

[23] J. Imola, T. Murakami, and K. Chaudhuri, "Locally differentially private analysis of graph statistics," 2020, https://arxiv.org/abs/2010.08688.

[24] Z. Liu, L. Huang, and H. Xu, "PrivAG: Analyzing attributed graph data with local differential privacy," in *Proceedings of the 26th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 422–429, Hong Kong, China, December 2020.

[25] H. Sun, X. Xiao, and I. Khalil, "Analyzing subgraph statistics from extended local views with decentralized differential privacy," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 703–717, London, UK, November 2019.

[26] J. Yang, X. Ma, X. Bai, and L. Cui, "Graph publishing with local differential privacy for hierarchical social networks," in *Proceedings of the 2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pp. 123–126, Beijing, China, July 2020.

[27] Q. Ye, H. Hu, and M. H. Au, "Towards locally differentially private generic graph metric estimation," in *Proceedings of the 36th IEEE International Conference on Data Engineering (ICDE)*, pp. 1922–1925, Dallas, USA, April 2020.

[28] Y. Zhang, J. Wei, and X. Zhang, "A two-phase algorithm for generating synthetic graph under local differential privacy," in *Proceedings of the Proceedings of the 8th International Conference on Communication and Network Security*, pp. 84–89, Qingdao, China, November 2018.

[29] M. Hay, C. Li, G. Miklau, and D. D. Jensen, "Accurate estimation of the degree distribution of private networks," in *Proceedings of the Ninth IEEE Inter - National Conference on Data Mining (ICDM)*, pp. 169–178, Miami, USA, December 2009.

[30] Y. Wang, X. Wu, J. Zhu, and Y. Xiang, "On learning cluster coefficient of private networks," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 925–938, 2013.

[31] X. Li, J. Yang, Z. Sun, and J. Zhang, "Differential privacy for edge weights in social networks," *Security and Communication Networks*, vol. 2017, p. 10, Article ID 4267921, 2017.

[32] A. Sala, X. Zhao, and C. Wilson, "Sharing graphs using differentially private graph models," in *Proceedings of the 11th ACM SIGCOMM Internet Measurement Conference*, pp. 81–98, Berlin, Germany, November 2011.

[33] W. Aiello, F. R. K. Chung, and L. Lu, "A random graph model for massive graphs," in *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pp. 171–180, Portland, USA, May 2000.

[34] J. J. Pfeiffer, S. Moreno, T. L. Fond, J. Neville, and B. Gallagher, "Attributed graph models: modeling network structure with correlated attributes," in *Proceedings of the 23rd International World Wide Web Conference (WWW)*, pp. 831–842, Seoul, Republic of Korea, April 2014.

[35] F. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," *Communications of the ACM*, vol. 53, no. 9, pp. 89–97, 2010.

[36] W. L. Croft, J. Sack, and W. Shi, "Differential privacy via a truncated and normalized laplace mechanism," *Journal of Computer Science and Technology*, 2019, https://arxiv.org/abs/1911.00602.

[37] J. J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks," in *Proceedings of the Advances in Neural Information Processing Systems 25 (NIPS 2012)*, pp. 548–556, Lake Tahoe, USA, December 2012.

[38] B. Rozemberczki, R. Davies, R. Sarkar, and C. Sutton, "GEMSEC: graph embedding with self-clustering," in *Proceedings of the 2019 International Conference on Advances in Social Networks Analysis and Mining*, pp. 65–72, Vancouver, Canada, August 2019.