Practical Privacy-Preserving Community Detection in Decentralized Weighted Networks

Tingxuan Han, Wei Tong(✉), Jiacheng Niu, and Sheng Zhong

State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
weitong@outlook.com

Abstract. As one of the essential graph analysis tasks, community detection plays a crucial role in various applications. Since a graph can be viewed as a collection of users' personal information and the relationships among them, it is essential to protect individuals' privacy during the process of community detection. Weighted networks are networks with real-valued weights on edges, and the edge weights should be protected when performing community detection in weighted networks. In this paper, we study the privacy-preserving community detection problem for weighted networks where each node is a distributed user and there is no trusted third party. Directly applying homomorphic encryption to implement a secure version of community detection for the untrusted analyst will incur high communication and computational cost, while local perturbation methods, such as local differential privacy, may significantly degrade the accuracy. Therefore, we combine these two types of techniques and propose a practical privacy-preserving algorithm for community detection in weighted networks, by blending a tailored locally differentially private mechanism with a cryptographic component. The proposed method can provide provable privacy guarantees and satisfactory performance. Extensive experiments we conducted have demonstrated the accuracy and efficiency of the proposed algorithm in various cases.

Keywords: Community detection · Decentralized weighted networks · Privacy-preserving · Local differential privacy

1 Introduction

Community detection is a fundamental task in graph analytics. It enables us to uncover valuable hidden information among vertices in the network. The key idea of community detection is to assign vertices in the network to groups, i.e., communities, such that the connection between vertices in the same community is closer than other parts of the network. Weighted networks are networks where edges are assigned weights and have applications in various areas, e.g., communication networks [10], weighted social networks [13], user-item graphs [7] and

H. Duan et al. (Eds.): SecureComm 2023, LNICST 568, pp. 302–320, 2025.
https://doi.org/10.1007/978-3-031-64954-7_16

scientific collaboration networks [15]. Weighted networks provide a richer representation of the relationships between vertices, and it is crucial to analyze these networks by performing community detection in order to obtain a more meaningful characterization in real-world applications.

However, the graphs may contain private information about individuals, such as their age, gender, location, and the presence or absence of edges indicates whether two users are friends. The weight on an edge can signify the strength of the relationship between two users in the social network or the rating of a product from a consumer in a user-item graph. Leaking such sensitive information can lead to serious privacy breaches and cause users to be more concerned about the protection of their data. Therefore, it is crucial to protect privacy while detecting communities in these graphs.

Existing works that preserve privacy for community detection often require a trusted central server, for applying differential privacy [17,22]. It could be a strong assumption that the analyst performing the community detection is trusted in a real-world application. If the cryptographic techniques, e.g., homomorphic encryption [8], are applied to handle the untrusted server, the intensive computation may be unaffordable for large-scale networks. Local differential privacy (LDP) has become a viable lightweight option that allows users to perturb their data locally before submitting it to the central server, thereby preventing the server from accessing the real data. Some recent works [21,26,29] have shown the potential of local differential privacy in the field of graph data analysis. These works demonstrate that the collection of graph topological structure and graph node attribute can be achieved under the LDP protection, but they may not apply well for community detection in weighted networks.

We study privacy-preserving community detection in weighted networks without assuming a trusted central server. The challenge of preserving the privacy of edge weights is twofold. On the one hand, in a decentralized network, each user can only observe the graph that is visible from its own local view, i.e., each user only knows its neighboring users and the weights on the edges to/from it, but the community detection may require global information such that the optimization objective can be fulfilled on the whole graph. On the other hand, directly applying LDP mechanisms solely to edge weights may severely affect accuracy due to the perturbation caused by the LDP mechanisms. In addition, as mentioned above, using cryptographic methods throughout the community detection process can preserve accuracy, but may result in heavy computational costs.

In this paper, we propose a **P**ractical **P**rivacy-**P**reserving **C**ommunity **D**etection algorithm in **W**eighted networks, named P^{3}CDW. Based on the fact that LDP mechanisms may not perform satisfactorily for selection problems but are effective for aggregation tasks, we identify which parts should be protected by LDP mechanisms or cryptographic methods by analyzing the key components, namely *seed selection* and *community expansion*, of a widely-used community detection algorithm [14]. Specifically, we leverage the cryptographic primitive that can find the maximum value and adapt it to a multi-round process of the

community detection algorithm with a well-designed caching strategy for the seed selection steps. By incorporating optimized hyperparameter selection into a tailored perturbation mechanism with weight truncation and discretization, we have designed an LDP mechanism to protect the edge weights for the community expansion steps.

The contributions of this paper are summarized as follows:

- To the best of our knowledge, we are the first to study the problem of privacy-preserving community detection in weighted networks with decentralized users and an untrusted analyst.
- By marrying LDP mechanisms and cryptographic techniques, we have proposed a practical algorithm for privacy-preserving community detection. In addition, our algorithm inherits the desirable features of the non-private algorithm [14] and can also detect overlapping communities.
- We have conducted extensive experiments to evaluate the proposed method, and the results demonstrate its accuracy and efficiency.

The remaining sections of this paper are organized as follows. Section 2 introduces the preliminaries of this paper. Section 3 formulates the problem and analyzes a naive solution. Section 4 introduces the proposed algorithm P^3CDW in detail. Section 5 presents experiments and analysis. Section 6 introduces the related work. Section 7 concludes this paper.

2 Preliminaries

2.1 Community Detection in Weighted Graphs

Suppose that there is an undirected and weighted graph $G = (V, E)$, where V is the set of vertexes (users) and E is the set of edges. Denoted by $(u, v) \in E$ the edge between two users u and v. Let W be the weight matrix of the graph, and $w_{uv} \geq 0$ is the weight of the edge (u, v). The degree of vertex u is denoted by d_u; the set of neighbors of vertex u is denoted by N_u. The task of community detection is to generate a set of clusters $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$, each cluster is a subset of V and $\bigcup_{i=1}^{K} C_i \subseteq V$. Let N_{C_i} be the set of neighboring vertexes of cluster C_i, *i.e.*, $N_{C_i} = \{v | (u, v) \in E, \exists u \in C_i, \forall v \in (V - C_i)\}$. If $C_i \cap C_j = \emptyset$ for any i, j, then the community detection is non-overlapping; otherwise, it is an overlapping community detection.

Typical community detection methods for weighted graphs in a centralized setting include spectral clustering [23], Louvain algorithm [1], and conductance-based method [14]. The first two methods may either lead to explosive computational complexity or be unsuitable for detecting overlapping communities. Because non-overlapping community detection can be achieved by overlapping community detection with straightforward post-processing. Therefore, we consider a more general setting for providing privacy preservation for overlapping community detection in this paper. We will focus on a well-known variant of the conductance-based community detection algorithm proposed by Lu et al. [14], which we will refer to as *Conductance* for simplicity.

Below we review two major concepts for conductance-based methods [14]:

– **Belongingness.** For a vertex u and a community C, the belongingness is defined as

$$B(u, C) = \frac{\sum_{v \in C} w_{uv}}{\sum_{l \in N_u} w_{ul}}. \tag{1}$$

– **Conductance.** For a community C, the conductance is defined as

$$\Phi(C) = \frac{\sum_{i \in C, j \notin C} w_{ij}}{\sum_{i,j \in C} w_{ij}}. \tag{2}$$

Lu et al.'s *Conductance* algorithm performs in an expansive way. It finds the edge with the largest weight in the graph and selects its endpoints as the initial community C_i^0. For each round t, the algorithm finds the vertex that has the largest belongingness with the current community:

$$u_t^* = \arg \max_{u \in N_{C_i^t}} B(u, C_i^t), \tag{3}$$

and set $C_i^{t+1} = C_i^t \cup \{u_t^*\}$. If $\Phi(C_i^{t+1}) < \Phi(C_i^t)$, the expanding process continues; otherwise, add C_i^t into $\mathcal{C}$ and end the expanding process. After that, the edges included by detected community C_i^t will be deleted from the graph G. The expansion process ends when the edge set E is empty.

2.2 Local Differential Privacy

Local differential privacy (LDP) [3] extends the standard (centralized) differential privacy (DP) [4] notion to the local model. Since the data is perturbed locally by each user, local differential privacy does not require a reliable central server to aggregate the data.

Definition 1 (Local Differential Privacy). *A randomized mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ satisfies (ϵ, δ)-local differential privacy, if and only if for any two inputs x and x' and for any output y of $\mathcal{M}$,*

$$\Pr[\mathcal{M}(x) = y] \leq e^{\epsilon} \Pr[\mathcal{M}(x') = y] + \delta.$$

Definition 2 (Sequential Composition). *Given h random mechanisms $\mathcal{M}_i (1 \leq i \leq h)$. If each $\mathcal{M}_i$ satisfied ϵ_i-LDP, then the composition mechanism $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_h\}$ satisfied $\sum_{i=1}^{h} \epsilon_i$-LDP.*

Based on the type of data, *i.e.*, categorical data or numerical data, to be protected, LDP mechanisms can be classified into two categories: Generalized Randomized Response (GRR) [24] is designed to protect categorical data, while Laplace Mechanism (LM) [5] and Harmony Mechanism (HM) [19] aim to protect numerical data. Denote by x the original value, y the perturbation value, and ϵ the privacy parameter. We review these three mechanisms below.

GRR. Let d be the size of the domain of the categorical data x, the mechanism perturbs the input with respect to the following probabilities:

$$\Pr[\mathsf{Perturb}_{\mathrm{GRR}}(x) = y] = \begin{cases} \frac{e^{\epsilon}}{e^{\epsilon}+d-1} & , y = x \\ \frac{1}{e^{\epsilon}+d-1} & , y \neq x \end{cases} \tag{4}$$

LM. Assume that the sensitivity is Δf, which would be the size of the domain of the numerical data x for the local setting, the Laplace Mechanism perturbs the value with the additive noise:

$$\mathsf{Perturb}_{\mathrm{LM}}(x) = x + Lap(\frac{\Delta f}{\epsilon}) \tag{5}$$

HM. The Harmony Mechanism perturbs the input to two real values with respect to the following probabilities:

$$\Pr[\mathsf{Perturb}_{\mathrm{HM}}(x) = y] = \begin{cases} \frac{x(e^{\epsilon}-1)+e^{\epsilon}+1}{2e^{\epsilon}+2} & , y = \frac{e^{\epsilon}+1}{e^{\epsilon}-1} \\ \frac{-x(e^{\epsilon}-1)+e^{\epsilon}+1}{2e^{\epsilon}+2} & , y = -\frac{e^{\epsilon}+1}{e^{\epsilon}-1} \end{cases} \tag{6}$$

2.3 Privacy-Preserving Selection

The privacy-preserving selection allows a semi-honest central server to securely compute the maximum or minimum value from a set of values, where each value is possessed by a separate user, while ensuring that no additional knowledge is revealed to any party. In this paper we adopt a construction of privacy-preserving selection proposed by Zhang et al. [30] as a primitive in our algorithm. For the sake of completeness, we review the major components of the construction below.

The main idea of Zhang et al.'s protocol is based on a bitwise XOR homomorphic encryption. Denote by $h_{s,\alpha,\beta} : \{0,1\}^{\alpha} \to \{0,1\}^{\beta}$ a pseudo-random function and $t \in \{0, 1, \ldots, 2^f - 1\}$ a random nonce.

- *Key generation.* The protocol independently generates $s_1, s_2, \ldots, s_n \in \{0,1\}^{\gamma}$ that confirm uniform distribution. Then each user u_i computes its secret key as
$$k_i = h_{s_i,f,l}(t) \oplus h_{s_{(i \bmod n)+1},f,l}(t).$$
- *XOR homomorphism.* For a bit string $x_i \in \{0,1\}^l$, user u_i encrypts/decrypts it by
$$\overline{x_i} = x_i \oplus k_i.$$
It is easy to verify that the bitwise XOR of all users' ciphertexts is equal to the bitwise XOR of all users' plaintexts:
$$x_1 \oplus \cdots \oplus x_n = \overline{x_1} \oplus \cdots \oplus \overline{x_n}.$$
- *Selection.* The protocol proposed by Zhang et al. only focuses on the selection of minimum value, we adapt it to fit the selection of maximum value. Each user u_i reports the ciphertext about its k^{th} bit to the analyst from the most

significant bit (MSB) to the least significant bit (LSB). In the beginning, all users are tagged as "affirmative". At the k^{th} round, u_i generates a string x_{ik} as follows:

$$x_{ik} = \begin{cases} C(u_i, k) & , u_i \text{ is "affirmative"} \\ 0^q & , u_i \text{ is "negative"} \end{cases} \tag{7}$$

The probabilistic coding scheme $C(u_i, k)$ is defined as follows:

$$C(u_i, k) \begin{cases} \overset{\$}{=} \{0,1\}^q & , k^{th} \text{ bit of } u_i \text{ is } 1 \\ = 0^q & , k^{th} \text{ bit of } u_i \text{ is } 0 \end{cases} \tag{8}$$

where q is the length of the result each user reports to the analyst, and $\overset{\$}{=}$ means each bit of the result conforms to a uniform distribution. Then u_i reports $\overline{x_{ik}} = x_{ik} \oplus k_i$ to the analyst. After receiving reports from all users, the analyst computes $x_i = \overline{x_{i1}} \oplus \overline{x_{i2}} \oplus \cdots \oplus \overline{x_{in}}$. If $x_i \neq 0^q$, then the k^{th} bit of the maximum value is 1; otherwise the k^{th} bit of the maximum value is 0. Then the analysts sends the k^{th} bit of the maximum value to all users, the user whose k^{th} bit is different from the maximum value changes its tag to "negative".

2.4 Threat Model

We assume that all participants are semi-honest, *i.e.*, the analyst and all users comply with the protocol but may attempt to infer information they are not supposed to know from the information they have obtained.

For each user, we assume that it knows which nodes are its neighbors and the weights of edges that connect to it, e.g., users only have information about the degree of intimacy between themselves and their friends, but they do not have any information about the relationships between other users.

We assume that the analyst knows the topology of the network, but does not have the knowledge of the weights on edges. This is a reasonable assumption for many real-world cases. For instance, a social network service provider has information about the friendship connections of users (topology), but it does not have any knowledge about the degrees of intimacy between users (weights). Additionally, for the user-item graph in shopping behavior analysis, the analyst knows the purchase history of users (topology), but collecting ratings from users to items (weights) requires additional data collection from users.

3 Problem Formulation and Straightforward Solutions

3.1 Problem Formulation

We first present the formulation of the privacy-preserving community detection problem in decentralized weighted networks.

Considering a decentralized weighted network G, in which each node is a distributed user, and the set of users is $U = \{u_1, u_2, \ldots, u_n\}$. Each user

u_i only has the partial graph information from its view, and that information, i.e., the neighbor weight list, can be modeled as a n-dimension vector $\boldsymbol{w}_i = [w_{i1}, w_{i2}, \ldots, w_{in}]$. Then from the global perspective, the graph can be represented as $G = (V, E, W)$, where $V = U$ and $W = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_n]^T$. The goal of the analyst is similar to that we have described in Subsec. 2.1. The analyst wants to detect communities with better structure with the power of edge weights, but it is not allowed to access the raw data of any $\boldsymbol{w}_i$.

3.2 Straightforward Solutions

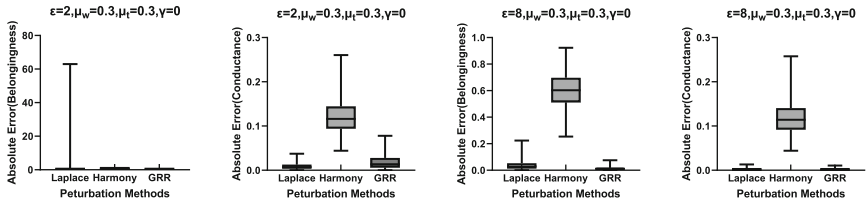
Instead of using a computationally expensive cryptographic method, a simpler approach to preserving user privacy during community detection is to apply a local differential privacy mechanism $\mathcal{M}$ to the edge weights before they are sent to the analyst. The process is shown as follows:

- Each user u_i perturbs its weight vector from $\boldsymbol{w}_i = \{w_{i1}, w_{i2}, \ldots, w_{in}\}$ to $\boldsymbol{w}'_i = \{\mathcal{M}(w_{i1}), \mathcal{M}(w_{i2}), \ldots, \mathcal{M}(w_{in})\}$, and then reports $\boldsymbol{w}'_i$ to the analyst.
- For round t, the analyst randomly selects an edge (u_t, v_t) from the graph and set the cluster as $C_i^t = \{u_t, v_t\}$. Then the analyst computes the expansion criteria based on the perturbed weight matrix $W' = [\boldsymbol{w}'_1, \boldsymbol{w}'_2, \ldots, \boldsymbol{w}'_n]^T$ from the users.

The choice of an appropriate LDP mechanism $\mathcal{M}$ is dependent on the characteristics of the target task, particularly the type of data that needs to be locally perturbed. The following provides a brief analysis of whether some of the classic local differential privacy mechanisms can be applied to the task of community detection in a weighted graph or edge weights, which are real-valued data, they can be directly perturbed by mechanisms that add additive noise, such as the Laplace Mechanism (LM) [5], or local randomization mechanisms designed for numerical values, such as the Harmony Mechanism (HM) [20]. However, if we want to apply LDP mechanisms to categorical data, such as the Generalized Randomized Response (GRR) [24], it is necessary to first apply discretization encoding before the perturbation.

Infeasibility of LM and HM. The Laplace Mechanism requires adding noise with respect to the sensitivity of the query, and a larger sensitivity means a larger scale of noise needs to be added to the output. In the case of Local Differential Privacy (LDP), the output of the query is identical to the input, and thus the sensitivity equals the range of the edge weights, which could be very large, or even unbounded.

The Harmony Mechanism is designed to perturb a weight by mapping it from a continuous range to binary encoding, with symbols $\{\frac{w(e^\epsilon-1)+e^\epsilon+1}{2e^\epsilon+2}, \frac{-w(e^\epsilon-1)+e^\epsilon+1}{2e^\epsilon+2}\}$. However, a large cohort of users is required to achieve the expected accuracy with this mechanism. At the beginning of the community detection process, only a small number of users and edges are involved due to the community not having expanded to a sufficiently large size. This



(a)Belongingness(ϵ=2) (b) Conductance(ϵ=2) (c)Belongingness(ϵ=8) (d)Conductance(ϵ=8)

Fig. 1. Absolute error of different perturbation methods on Belongingness and Conductance

makes the Harmony Mechanism face a cold start and results in a disastrous performance of community detection.

We conducted a micro-experiment involving 1,000 communities, each comprising 50 members, to calculate the values of Belongingness and Conductance with different perturbation methods for each community. The result shown in Fig. 1 indicates that LM and HM exhibit significant instability in their computed results due to inherent flaws in these methods. Therefore, we will discuss the perturbation scheme based on GRR in the next part.

GRR-Based Solution. Due to the infeasibility of LM and HM, we choose Generalized Randomized Response (GRR) as the perturbation mechanism for achieving local differential privacy in the straightforward solution. Firstly, the analyst determines the number of buckets d, and then each user u_i assigns non-zero weights to corresponding buckets. The represent value of bucket i is $\frac{2i-1}{2}$. After that, u_i perturbs each non-zero edge's bucket id by GRR and reports the corresponding perturbed value to the analyst. Finally, the analyst conducts a community detection algorithm based on the reported perturbed weights.

For community detection, the major drawback of GRR is that its randomization process outputs values in a different domain, and the probability of a weight being perturbed to any other value is the same. This property can result in perturbed values that can deviate significantly from the true value, making it difficult to compare two values based on the perturbed values. In addition, GRR also faces the challenge of how to select parameters of discretization. On the one hand, using a coarse-grained strategy for discretization may result in significant information loss. On the other hand, if we use a large number of buckets to discretize the weights, the perturbation introduced by GRR may overwhelm the true value. This is because the accuracy of GRR depends on the size of the output domain.

4 P^{3}CDW

4.1 Overview

We have a key observation that typical community detection algorithms in weighted networks mainly consist of two types of computation: maximum value

selection and mean aggregation. The major idea of P^{3}CDW is to combine homomorphic encryption with local differential privacy together to design a practical privacy-preserving community detection algorithm based on the *Conductance* algorithm [14]. Specifically, we exploit the advantage of the LDP mechanism that it can execute efficiently while providing privacy for each user against an untrusted analyst, and use it to secure the mean aggregation. For the selection operations in the community detection, we use a privacy-preserving selection technique, as mentioned in Subsec. 2.3.

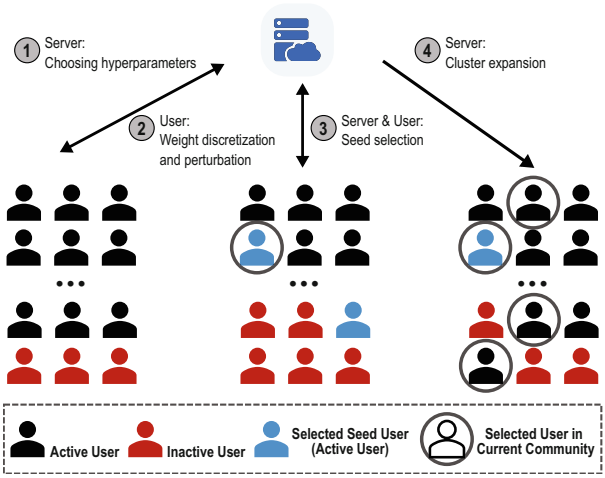


Fig. 2. Overview of P^{3}CDW

The major steps of P^{3}CDW describe as follows:

- **Choosing hyperparameters.** We use the GRR mechanism to perturb the edge weights. The analyst determines the key hyperparameters: a clipping threshold t, the number of categories d used for discretization, and the privacy budget ϵ. These hyperparameters will be shared among all the users.
- **Weights discretization and perturbation.** Each user truncates and discretizes the weights on the adjacent edges according to t and d, respectively, perturbs each non-zero weight by GRR with a privacy budget of ϵ, and sends the perturbed weights to the analyst. We will provide more details on this step in Subsec. 4.2.
- **Seed selection.** As part of the community detection process, the analyst selects a seed user u_{seed} for each new community using a privacy-preserving seed selection protocol, which is introduced in Sect. 4.3. This ensures that the community detection algorithm is initialized in a privacy-preserving manner with untrusted analyst.
- **Cluster expansion.** After the analyst obtains the information of the seed user u_{seed} through the seed selection process, the initial community consists

of only this user, i.e., $C = u_{\text{seed}}$. The analyst enumerates all the neighboring nodes of C and adds the neighbor with the maximal belongingness $B(u, C)$ to the community, resulting in C'. The analyst repeatedly expands this community until its conductance $\Phi(C') \geq \Phi(C)$ in any round. At the end of each round, the edges within the community C are removed from the set of edges. The belongingness and conductance are computed based on perturbed weightst to ensure privacy.

4.2 Weights Discretization and Perturbation

In order to satisfy the input format of GRR, the edge weights need to be converted from real value to category data. Our approach divides the possible value range into d equally-sized parts, each of which is represented by the middle value of the interval. For example, if the value range is $[0, 1000]$ and $d = 10$, then the origin weight 173.61 corresponds to the category 2 whose represent value is 150.

In many real-world cases, such as social networks, edge weights obey the power law distribution which is a type of long-tail distribution. Therefore, if the weights are converted to categorical data directly, there will be few items in categories that correspond to large numerical values. To solve this problem, any weight that is greater than the threshold value t is truncated to t before being converted to categorical data. For example, if the maximal weight value is 1000 and about 10 percent of weights are greater than 500, then the range of weights can be truncated to $[0, 500]$. Here, we use the 90th percentile as the threshold and reduce the number of categories by half. Perturbing values into a smaller domain, according to the properties of GRR, can result in higher accuracy with a fixed privacy budget.

If each user applies GRR to perturb its non-zero edges, then the weight on each non-zero edge will be unnecessarily perturbed twice, which means that if each time the privacy budget consumed is ϵ, then this method satisfies 2ϵ-LDP instead of ϵ-LDP. To deal with this issue, inspired by the method used by Ye et al. [29] to avoid duplicate reports for topology statistic collection, we let each user only perturbs at most $\lfloor \frac{n}{2} \rfloor$ edges for the network with n users. Specifically, user i only perturbs the non-zero edges with the users whose number is in $\{(i + k)\%n + 1 | k = 1, 2, \cdots, \lfloor \frac{n}{2} \rfloor\}$. Then each weight on the non-zero edge will be perturbed only once.

Theorem 1. *Weights discretization and perturbation satisfies ϵ-LDP.*

Proof. Since the weight on each edge is perturbed one time by GRR with privacy budget ϵ, and GRR satisfies ϵ-LDP. Meanwhile, it is obvious that weight discretization does not violate user privacy. Hence it is clear that the weight discretization and perturbation process satisfies ϵ-LDP. □

Optimizing the Hyperparameters. There are two sources of error in this process: discretization and perturbation. On the one hand, fine-grained discretization requires a large d and a large t, which can provide a better approximation of the original values. On the other hand, a large threshold and a large

number of categories can result in a large output domain for the GRR mechanism, leading to an unacceptable error in the perturbation. Below, we attempt to formulate the error caused by weight discretization and perturbation while considering two important hyperparameters: the number of categories d and the threshold value t.

Assuming that the probability density function of the weight distribution is $p(x)$, the range of weight is $[0, m]$, then our goal is shown as follows:

$$\arg\min_{d,t} \sum_{i=0}^{d-1} \int_{\frac{it}{d}}^{\frac{(i+1)t}{d}} p(x)|x - \frac{(2i+1)t}{2}|\mathrm{d}x + \int_{t}^{m} p(x)(x - \frac{(2d-1)t}{2})\mathrm{d}x$$

Due to the requirement of GRR, weights that are no larger than t are projected to their corresponding representing values. Considering the distribution of the data, the error caused by this projection is given by $\sum_{i=0}^{d-1} \int_{\frac{it}{d}}^{\frac{(i+1)t}{d}} p(x)|x - \frac{(2i+1)t}{2}|\mathrm{d}x$. After truncation, the range of weight is $[0, t]$, and all weights larger than t are truncated to t. Therefore, the representing value of these weights is $\frac{(2d-1)t}{2}$, so the error of this part is $\int_{t}^{m} p(x)(x - \frac{(2d-1)t}{2})\mathrm{d}x$. Because d is an integer and t is large enough to ignore the influence of small errors, it can be viewed as an integer. Thus, a basic method to get a second-best solution is to traverse the search space of d and t and determine the hyperparameters.

4.3 Privacy-Preserving Seed Selection

The strength of LDP lies in analyzing aggregated data, such as frequency, mean value, and so on, but it cannot be applied well to selection tasks. However, selecting the seed edge (*i.e.*, the edge with the largest weight of the remaining edges) precisely is crucial for community detection. To address this problem, P^{3}CDW applies a cryptographic component to select the edge with the largest weight from a set of edges without revealing the weights to an untrusted analyst.

Inspired by the idea introduced by Zhang et al. [30], we propose a novel cryptographic component for community detection to select the seed user. In addition, we leverage the structure of the community detection algorithm to further reduce the number of secure comparison operations required. At the beginning of P^{3}CDW, each user chooses the largest weight edge connecting itself with the other user whose *id* is smaller than its as representing weight, and then each user sets its status as "Active". When selecting the seed user for a new community, the algorithm considers only the users whose status is currently "Active". During this stage, involved users will compare each bit of their representing weight from the most significant bit (MSB) to the least significant bit (LSB) to determine the user whose weight is the largest. At the comparison of the k-th bit, each active user will report their k-th bit using the probabilistic coding scheme introduced in Subsec. 2.3.

To protect each user's privacy, each user uses the cipher introduced in Subsec. 2.3 to encrypt its report. Once the analyst receives reports from participating users, it computes the bitwise XOR of the encrypted reports. By leveraging the

encryption method described in Subsec. 2.3 and the properties of XOR, it is easy to see that the XOR result computed by the analyst is equivalent to the XOR result of all plaintext reports. If a user's k-th bit is 0, then its plaintext report is 0^q, and its ciphertext report will not change; otherwise, the probability that two consecutive ciphertexts are the same is $\frac{1}{2^q}$. Hence, the analyst can infer a user's k-th bit is 0, if its ciphertexts for $(k-1)$-th and k-th are the same. If the k-th bit is inferred to be 0 for all participating users or the XOR result of all ciphertext reports is 0^q, the algorithm continues to compare the next bit. If only one user's k-th bit is inferred to 1, the algorithm selects this user as the seed of the community and stops the comparison process. Otherwise, the status of each user whose k^{th} bit is inferred to 0 will be changed to "Inactive"(Fig. 3).

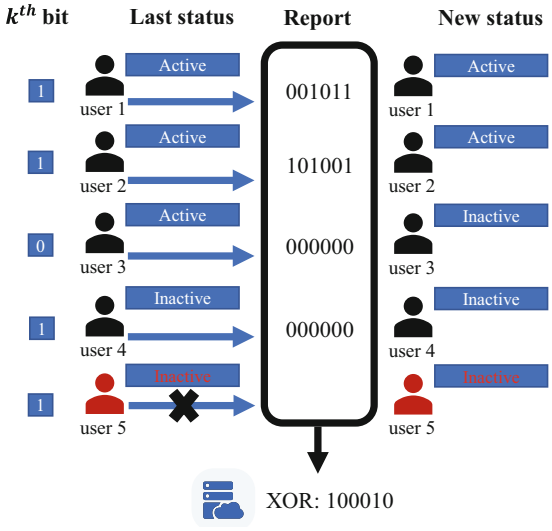


Fig. 3. Users whose status is "Inactive" before will not participate in seed edge selection. If a user's status is from "Active" to "Inactive" during the selection, it will report 0^q after the encryption of the cipher introduced in Subsec. 2.3 until the selection stops

After the current community $\mathcal{C}$ is detected, all users assigned to this community will update their representing weights and then change their status to "Active". For a user $u_i \in \mathcal{C}$, its new representing weight is the largest weight among edges connecting to the other user whose id is less than its and has not been assigned to any community.

The limitation of participants in each round of seed user selection and the stop mechanism of comparison in each round can reduce the computation significantly, which will be demonstrated by extensive experiments in Subsec. 5.3.

Theorem 2. *Seed user selection is perfectly privacy-preserving for all participants.*

Table 1. Parameter settings in Benchmark.

Parameter	Value	Meaning
N	5000	Number of nodes
μ_w	0.1,0.3,0.5	Mixing parameter for edge weights
μ_t	0.1,0.3,0.5	Mixing parameter for topology
β	2	Exponent for weight distribution
k	50	Average node degree
$k_{\max}$	100	Maximum node degree
t_1	2	Minus exponent for degree sequence
t_2	2	Minus exponent for community size distribution
γ	0.2, 0.3, 0.4	Overlapping fraction
o_m	2	Number of communities of the overlapping nodes

Proof. During the process of seed user selection, each user only reports the ciphertext to the analyst, so according to the proof in [30], this is perfectly privacy-preserving. Then the inference that whether a user's status will be changed to "Inactive" is based on the ciphertexts rather than plaintexts, so this does not violate the user's privacy. Therefore, seed user selection is perfectly privacy-preserving for all participants. □

5 Evaluation

5.1 Setup

Datasets. We implement P^{3}CDW and use the benchmark graph generate algorithm proposed by Lancichinetti et al. [11] to evaluate it. Weights in benchmark graphs generated by this method obey power law distribution, which can better simulate the weighted graphs in real-world cases. We list the parameters to generate the benchmark graphs in Table 1. The number of nodes in the graph is denoted as N and γN is the number of overlapping nodes which belong to more than one community. If the community detection is conducted on a graph with $\gamma \neq 0$, then it is called overlapping community detection. Overlapping nodes belong to at most o_m communities. The parameters k and $k_{\max}$ control the average and maximum node degree, respectively. The parameter β decides the weight distribution of generated graph. The other parameters, such as μ_w, μ_t, t_1, t_2, allow more variations of the benchmark graphs for more general experiments.

Considering that most social networks in the real world have overlapping communities, *e.g.*a person can be a member of both a baseball club and a football club while our goal of community detection is to find all members for each sport club. In addition, taking the changes of topology and weight structure in the real social graph into account, we evaluate the performance of P^{3}CDW with

nine weighted networks with the overlapping fraction $\gamma = 0.2$,0.3,0.4 and mixing parameters $\mu_w = \mu_t = 0.1$, $\mu_w = \mu_t = 0.3$ and $\mu_w = \mu_t = 0.5$, respectively.

Evaluation Metric. We use a widely-adopted metric, Normalized Mutual Information (NMI), to evaluate these methods, which is proposed by Lancichinetti et al. [12] as an evaluation metric to measure the similarity between two community structures.

Suppose $\mathbf{X}$ and $\mathbf{Y}$ are two community structures, their NMI can be calculated as follows:

$$\mathrm{NMI}(\mathbf{X}|\mathbf{Y}) = 1 - \frac{1}{2}[H(\mathbf{X}|\mathbf{Y})_{\mathrm{norm}} + H(\mathbf{Y}|\mathbf{X})_{\mathrm{norm}}] \tag{9}$$

Due to the normalization, the range of $\mathrm{NMI}(\mathbf{X}|\mathbf{Y})$ is $[0, 1]$. The higher $\mathrm{NMI}(\mathbf{X}|\mathbf{Y})$ indicates the community structure of $\mathbf{X}$ is more similar to $\mathbf{Y}$. Since $\mathbf{Y}$ is the baseline community structure, the more similar $\mathbf{X}$ is to $\mathbf{Y}$ means the better performance of $\mathbf{X}$.

Baselines. To verify the importance of weights in community detection, we ignore the weights in the graph, *i.e.*, if there is an edge between u_i, u_j, then $w_{ij} = 1$, otherwise $w_{ij} = 0$. Then we conduct the algorithm *Conductance* on this binary graph, and denote the community structure as Binary.

The straightforward solutions mentioned in Subsec. 3.2 can serve as a comparison to verify the performance of P^{3}CDW. Since Laplace brings too much noise for the numerical weights while Harmony loses too much information, which causes extremely poor performance of community detection in the beginning stage, the problems mentioned above make these two methods even can't complete the community detection. Therefore, we use the protocol of GRR as the comparison to confirm that P^{3}CDW performs better than some naive straightforward solutions.

5.2 Performance of P^{3}CDW

Figure 4 shows the performance of P^{3}CDW in nine kinds of benchmark graphs with overlapping fraction $\gamma = 0.2, 0.3, 0.4$ and mixing parameters $\mu_w = \mu_t = 0.1$, $\mu_w = \mu_t = 0.3$ and $\mu_w = \mu_t = 0.5$, respectively by comparing NMI with other methods. A smaller value of μ_w and μ_t indicates a tighter connection within the community. In this scenario, good community detection results can be achieved by utilizing only the topological information. However, with the value of μ_w and μ_t increasing, relying solely on topological information becomes challenging to obtain good community detection results. At this point, the advantage of leveraging weight information becomes evident. Hence, we can see that when $\mu_w = \mu_t = 0.5$, the performance of P^{3}CDW consistently performs better than the other two methods, while when $\mu_w = \mu_t = 0.1$, especially for smaller values of ϵ, P^{3}CDW may not outperform Binary that only makes use of topological information.

All line graphs in Fig. 4 show that with the increasing privacy budget ϵ, the performance of GRR and P^{3}CDW are better than Binary which loses the weight information totally. This indicates that utilizing weights is important

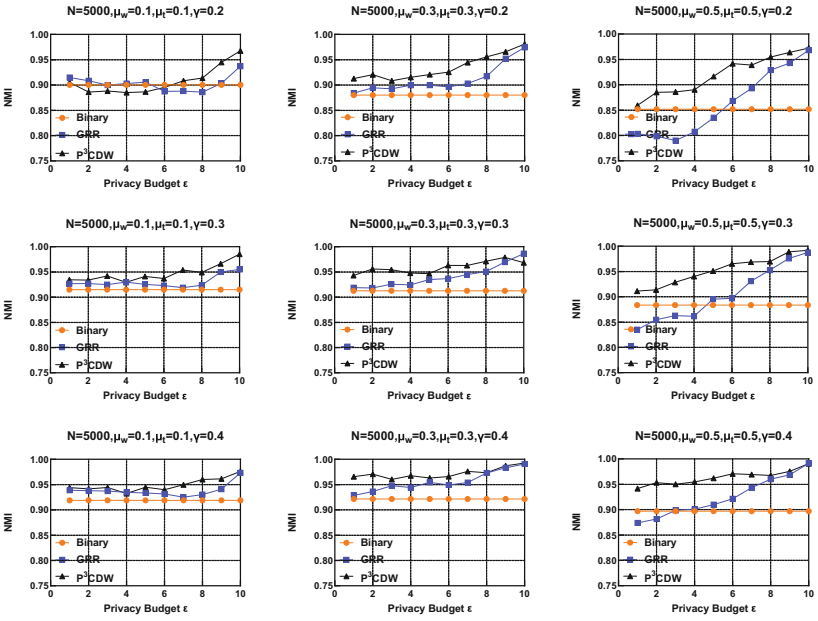


Fig. 4. The quality (*i.e.*NMI($\mathbf{X}|\mathbf{Y}$)) of the community structure detected by Binary, GRR and P^{3}CDW in nine kinds of weight graphs with different generating parameters

in community detection. Since the range of weight is large which means the sensitivity of weight is large, a small privacy budget ϵ will bring lots of noise that can't be neglected. Therefore, at the beginning of the line graphs, it is difficult to distinguish between the performance of these methods. However, it is clear that the performance of P^{3}CDW finally overcomes the other methods. Let's view the graphs in each column, as the overlapping fraction γ increases, P^{3}CDW performs better. This indicates that P^{3}CDW can adapt well to overlapping community detection.

Then we test P^{3}CDW with different scales of users. The generating parameter is $\mu_w = \mu_t = 0.3, \gamma = 0.2$, the number of users N ranges from 1,000 to 10,000, with an interval of 1,000. Two line graphs in Fig. 5 show the performance of P^{3}CDW with privacy budget $\epsilon = 2, 8$ respectively. It is clear that P^{3}CDW consistently outperforms the other two methods, which means that P^{3}CDW adapts different scales of users well.

5.3 Efficiency of Cryptographic Component

Considering the significance of seed user selection in community detection, P^{3}CDW decides to use some cryptographic methods to improve the accuracy of the selected user. Nevertheless, if we directly use the existing cryptographic methods, such as [30], it will bring a lot of computational overhead. Therefore,

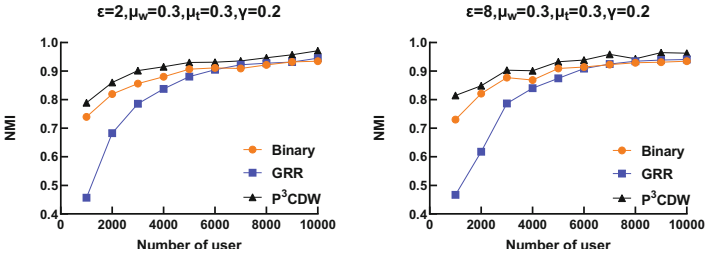


Fig. 5. The quality (*i.e.*NMI($\mathbf{X}|\mathbf{Y}$)) of the community structure detected by Binary, GRR and P^{3}CDW in different scales of users

combining with the actual situation of weighted social graphs, we propose a new cryptographic component that is more suitable for the scenario in this paper which can save a lot of computational sources.

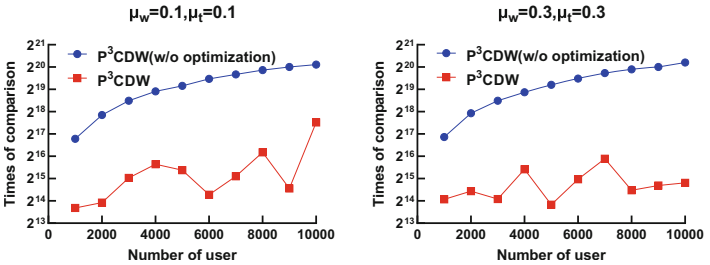


Fig. 6. Times of comparison cost by the cryptographic protocol used in P^{3}CDW with/without optimization

To demonstrate the efficiency of this new cryptographic component, we conducted experiments to calculate the numbers of comparison performed the cryptographic component used in P^{3}CDW with/without optimization with different mixing parameters for the number of nodes between 1,000 and 10,000. Fig. 6 shows that compared with the cryptographic component without optimization, the optimized cryptographic component saves a lot of computing resources effectively.

6 Related Work

For community detection, there are a lot of scholars researched and proposed many classic algorithms. Zhu et al. [28] proposed LPA algorithm which detects community by propagating labels on vertexes. The GN algorithm proposed by Newman and Girvan [16] constantly deletes the edge with largest betweenness which represents the degree of participation of edges until the stop standard is

reached. Blondel et al. proposed Louvain algorithm [1], which keeps merging communities into larger one until the modularity stops increasing. Recently, Lu et al. [14] proposed the *Conductance* algorithm and Whang et al. [27] proposed NISE algorithm. The former takes advantages of belonging degree to choose next vertex to be added in current community while the latter use Personalized PageRank algorithm to detect overlapping communities.

Local differential privacy(LDP) is a privacy model based on traditional differential privacy(DP) to deal with the situations where data is distributed in different users and there is no trustworthy data server. Existing LDP mechanisms are mainly applied on frequency estimation on categorical or numerical data. Firstly, Warner [25] proposed the classic algorithm randomized response. Then Duchi et al. [3] gave a upper error bound of LDP from the perspective of information theory. Hsu et al. [9] proposed a LDP mechanism to estimate heavy hitters by utilizing random projection and concentration. Erlingsson et al. [6] proposed Rappor, which is the first practical application of LDP mechanism for frequency estimation by Google.

Nowadays, LDP is also gradually used to protect graph data. Qin et al. [21] proposed LDPGen, which can generate valuable synthetic graphs under LDP by collecting the node degree of each user. Graphs generated by LDPGen can be used for a lot of graph analysis, such as statistical analysis of graph structure, social recommendation, etc. However, LDPGen doesn't take the attribute information into consideration. So Wei et al. [26] proposed AsgLDP to capture the attributes in graphs. In order to solve the problem of difficult implementation of LDP framework, Ye et al. [29] proposed a universal LDP framework LF-GDPR compatible with various graph analysis tasks.

The existing community detection researches under privacy protection can be roughly divided into three categories. Firstly, collecting data under privacy protection and then generating synthetic graphs to run existing community detection algorithms, such as LDPGen [21] and the probability graph proposed by Dev [2]. Secondly, collecting data to calculate unbiased atom information which is sufficient to conduct community detection algorithms is also a good choice, such as LF-GDPR [29]. The last one is to modify existing algorithms or propose new algorithms so that they can satisfy the requirements of privacy, such as LouvainDP proposed by Nguyen et al. [18] and LDPCD proposed by Zhang [31]. However, LouvainDP needs a trusted data server, while LDPCD only protects node degree so that it can't perform well on weighted graphs.

7 Conclusions

We study the problem of preserving the privacy of weights on edges with an untrusted analyst during the process of community detection in decentralized weighted networks. We have proposed an effective method P^3CDW, which combines a tailored local differential privacy algorithm with an efficient cryptographic component to improve the performance of community detection with privacy preservation. Extensive experiments on various benchmark graphs demonstrate the accuracy and efficiency of the proposed method.

Acknowledgment.. This work was supported in part by the National Natural Science Foundation of China under Grants NSFC-62002159, NSFC-62372226, NSFC-61872176, NSFC-62272215, the Jiangsu Shuangchuang Talent Program (No. JSSCBS20210035), the Leading-edge Technology Program of Jiangsu-NSF under Grant BK20222001 and BK20202001.

References

1. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theory Exp. **2008**(10), P10008 (2008)
2. Dev, H.: Privacy preserving social graphs for high precision community detection. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, SIGMOD 2014, pp. 1615–1616. Association for Computing Machinery, New York, NY, USA (2014)
3. Duchi, J.C., Jordan, M.I., Wainwright, M.J.: Local privacy and statistical minimax rates. In: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, pp. 429–438. IEEE (2013)
4. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
5. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
6. Erlingsson, Ú., Pihur, V., Korolova, A.: Rappor: Randomized aggregatable privacy-preserving ordinal response. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 1054–1067 (2014)
7. Feng, H., Tian, J., Wang, H.J., Li, M.: Personalized recommendations based on time-weighted overlapping community detection. Inf. Manage. **52**(7), 789–800 (2015)
8. Fontaine, C., Galand, F.: A survey of homomorphic encryption for nonspecialists. EURASIP J. Inf. Secur. **2007**, 1–10 (2007)
9. Hsu, J., Khanna, S., Roth, A.: Distributed private heavy hitters. In: Czumaj, A., Mehlhorn, K., Pitts, A., Wattenhofer, R. (eds.) ICALP 2012. LNCS, vol. 7391, pp. 461–472. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31594-7_39
10. Kossinets, G., Watts, D.J.: Empirical analysis of an evolving social network. Science **311**(5757), 88–90 (2006)
11. Lancichinetti, A., Fortunato, S.: Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Phys. Rev. E **80**(1), 016118 (2009)
12. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. New J. Phys. **11**(3), 033015 (2009)
13. Li, P., Yu, J., Liu, J., Zhou, D., Cao, B.: Generating weighted social networks using multigraph. Phys. A **539**, 122894 (2020)
14. Lu, Z., Sun, X., Wen, Y., Cao, G., Porta, T.L.: Algorithms and applications for community detection in weighted networks. IEEE Trans. Parallel Distrib. Syst. **26**(11), 2916–2926 (2015)

15. Menichetti, G., Remondini, D., Panzarasa, P., Mondragón, R.J., Bianconi, G.: Weighted multiplex networks. PloS one **9**(6), e97857 (2014)
16. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (2004)
17. Nguyen, H.H., Imine, A., Rusinowitch, M.: Detecting communities under differential privacy. In: Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society, pp. 83–93 (2016)
18. Nguyen, H.H., Imine, A., Rusinowitch, M.: Detecting communities under differential privacy. In: Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society, WPES 2016, pp. 83–93. Association for Computing Machinery, New York, NY, USA (2016)
19. Nguyên, T.T., Xiao, X., Yang, Y., Hui, S.C., Shin, H., Shin, J.: Collecting and analyzing data from smart device users with local differential privacy. arXiv preprint arXiv:1606.05053 (2016)
20. Nguyên, T.T., et al.: Collecting and analyzing data from smart device users with local differential privacy. CoRR **abs/1606.05053** (2016)
21. Qin, Z., Yu, T., Yang, Y., Khalil, I., Xiao, X., Ren, K.: Generating synthetic decentralized social graphs with local differential privacy. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 425–438 (2017)
22. Shao, Z., Ma, L., Lin, Q., Li, J., Gong, M., Nandi, A.K.: PMCDM: privacy-preserving multiresolution community detection in multiplex networks. Knowl.-Based Syst. **244**, 108542 (2022)
23. Von Luxburg, U.: A tutorial on spectral clustering. Stat. Comput. **17**(4), 395–416 (2007)
24. Wang, T., Blocki, J., Li, N., Jha, S.: Locally differentially private protocols for frequency estimation. In: 26th USENIX Security Symposium (USENIX Security 17), pp. 729–745 (2017)
25. Warner, S.L.: Randomized response: a survey technique for eliminating evasive answer bias. J. Am. Stat. Assoc. **60**(309), 63–69 (1965)
26. Wei, C., Ji, S., Liu, C., Chen, W., Wang, T.: AsgLDP: collecting and generating decentralized attributed graphs with local differential privacy. IEEE Trans. Inf. Forensics Secur. **15**, 3239–3254 (2020)
27. Whang, J.J., Gleich, D.F., Dhillon, I.S.: Overlapping community detection using neighborhood-inflated seed expansion. IEEE Trans. Knowl. Data Eng. **28**(5), 1272–1284 (2016)
28. Xiaojin, Z., Zoubin, G.: Learning from labeled and unlabeled data with label propagation. Technical Report, Technical Report CMU-CALD-02–107, Carnegie Mellon University (2002)
29. Ye, Q., Hu, H., Au, M.H., Meng, X., Xiao, X.: LF-GDPR: a framework for estimating graph metrics with local differential privacy. IEEE Trans. Knowl. Data Eng. **34**(10), 4905–4920 (2020)
30. Zhang, Y., Chen, Q., Zhong, S.: Efficient and privacy-preserving min and k th min computations in mobile sensing systems. IEEE Trans. Dependable Secure Comput. **14**(1), 9–21 (2015)
31. Zhang, Z.: LDPCD: a novel method for locally differentially private community detection. Comput. Intell. Neurosci. **2022**, 4080047 (2022)