Release Connection Fingerprints in Social Networks Using Personalized Differential Privacy

Yongkai Li*, Shubo Liu†, Jun Wang†, and Mengjun Liu*†
*School of Computer, Wuhan University, Wuhan, China
†Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, Wuhan University, Wuhan, China
Email: whu_lyk@foxmail.com, liu.shubo@whu.edu.cn, jameswang@whu.edu.cn, mengjun@hawaii.edu

Abstract—There are many benefits of publication social networks statistics for societal or commercial purposes, such as political advocacy and product recommendation. It is very challenging to protect the privacy of individuals in social networks while ensuring a high accuracy of the statistics. Moreover, most of the existing work on differentially private social network publication ignores the facts that different users may have different privacy preferences and there also exists a considerable amount of users whose identities are public. In this paper, we aim to release the number of public users that a private user connects to within $n$ hops (denoted as $n$-range Connection Fingerprints, or $n$-CFPs for short) regarding user-level personalized privacy preferences. To this end, we proposed two schemes, DEBA and DUBA-LF, for privacy-preserving publication of the CFPs on the base of personalized differential privacy (PDP), and conduct a theoretical analysis of the privacy guarantees provided within the proposed schemes. The implementation showed that the proposed schemes are superior in publication errors on real datasets.

I. INTRODUCTION

Nowadays, more and more people join multiple social networks on the Web, such as Facebook, Twitter, and Sina Weibo, to share their own information and at the same time to monitor or participate in different activities. Many institutions and firms are investing time and resources into analyzing social networks to address a diverse set of societally or commercially important issues including disease transmission, product recommendation, and political advocacy, among many others. Although the sophistication of information technology has allowed the collection, analysis, and dissemination of social network data, privacy concerns have posed significant restriction of the ability of social scientists and others to study these networks. To respect the privacy of individual participants in the networks, network data cannot be released for public access and scientific studies without proper sanitization.

A common practice is to release a “naively anonymized” isomorphic network after removing the real identities of vertices. It is now well-known that this can leave participants open to a range of attacks [1], [2]. Thus, a great many of anonymization techniques have been proposed [2], [3], [4], [5], [6] to ensure network data privacy. However, those anonymization techniques have been shown to be susceptible to several newly discovered privacy attacks and lack of rigorous privacy and utility guarantees. In response, differential privacy [7] has been applied to solve such vulnerability in social network data publication. Differential privacy is a popular statistical model, and it prevents any adversary from inferring individual information from the output of a computation by perturbing the data prior to the release. A limitation of the model is that the same level of privacy protection is afforded for all individuals. However, it is common that different users may have different privacy preferences [8], [9]. Therefore, providing the same level privacy protection to all the users may not be fair and in addition may cause the published social network data useless. Moreover, in reality, not all the identities of social network users are sensitive [6]. For instance, Sina Weibo, a popular Chinese microblogging social network, hosts around lots of media accounts (e.g., NBA and Xinhuanet), and millions of celebrity accounts (e.g., Christine Lagarde and Kai-Fu Lee). All these users identities are public, and they in total account for over 1% of the overall half billion registered user accounts [10].

In this paper, we classify the users whose identities are not sensitive as public users to be distinguished from private users. It is pointed out that releasing the identities of public users with social network data can benefit both research and the users themselves [6]. Moreover, we take different privacy requirements into account to guarantee precisely the required level of privacy to different users.

In this work, we focus on a specific publication goal when public users are labeled, i.e., the number of public users that a private user connects to within $n$ hops. For ease of presentation, we also use the definition $n$-Range connection fingerprints (CFPs) [6] to denote the public users that a private user connects to within $n$ hops. We choose to focus on the number of CFPs because it is one of the most important properties for a public users labeled graph. For example, these statistics can be used for studying the social influence of government organizations, simulating information propagation through media, helping corporate make smart targeted advertising plans, and so on.

In this work, we consider the setting in which a trusted data analyst desires to publish the number of $n$-Range CFPs of each private user. Every private user potentially requires different privacy guarantee for his or her statistics and the analyst would like to publish useful aggregate information about the network. To this end, we employ a new privacy framework, Personalized Differential Privacy (PDP) [9], to provide personal privacy guarantee specified at the user-level, rather than by a single,
global privacy parameter. The privacy guarantees of PDP have the same strength and attack resistance as differential privacy, but are personalized to the preferences of all users in the input domain.

In this work, we propose two schemes to release the number of CFPs in the context of personalized privacy preferences. We address the challenge of improving the data utility by employing the distance-based approximation mechanism and decreasing the introduced noise. The main contributions of this paper are:

1) To the best of our knowledge, we formalized the question of releasing the number of CFPs in the context of personalized privacy preferences for the first time.

2) We present two schemes, DEBA and DUBA-LF, for privacy-preserving publication of the CFPs regarding personalized privacy preferences, and we conduct a theoretical analysis of the privacy guarantees provided within the proposed schemes. The proposed schemes are designed to be \( \mathcal{P} \)-PDP.

3) We experimentally evaluate the two proposed schemes on real datasets and it is demonstrated that our proposed schemes have high utility for each dataset.

The paper is organized as follows. Section II discusses preliminaries and related work. Section III presents the problem statement and privacy goal. Overview of our solutions is described in Section IV. Section V presents our methods for privacy-preserving CFPs publishing. The privacy analysis is reported in Section VI. Section VII describes some of our experimental results and performance analysis. Section VIII presents the conclusions of this research.

II. PRELIMINARIES

In this section, we introduce some notations and initial definitions, and review the definition of differential privacy, two conventional mechanisms to achieve differential privacy, upon which our work is based. Then the related work is discussed.

We model a social network as an undirected and unweighted graph \( G = (V, E) \in \mathcal{G} \), where \( V \) is a set of vertices representing user entities in the social network, and \( E \) is a set of edges representing social connections between users (e.g., friendships, contacts, and collaborations). The notation \( e(v_i, v_j) \in E \) represents an edge between two vertices \( v_i \) and \( v_j \). We let \( |V| = n_0 \) and the notation \( |V| \) is used to represent the cardinality of \( V \). For ease of presentation, we use “graph” and “social network” interchangeably in the following discussion, as well as “user” and “node”.

A. Differential Privacy

We call two graphs \( G, G' \) as neighboring if \( G' \) can be obtained from \( G \) by removing or adding a one edge, i.e., their minimum edit distance [11] \( d(G, G') \leq 1 \). We write \( G \xrightarrow{e} G' \) to denote that \( G \) and \( G' \) are neighbors and that \( G = G' \wedge e \) or \( G' = G \wedge e \), where \( e \) is an edge. Differential privacy requires that, prior to \( f(G) \)'s release, it should be modified using a randomized algorithm \( \mathcal{A} \), such that the output of \( \mathcal{A} \) does not reveal much information about any edge in \( G \). The definition of differential privacy is shown as follows:

**Definition 1:** (\( \epsilon \)-differential privacy) [7] A randomized algorithm \( \mathcal{A} \) is \( \epsilon \)-differentially private if for any two graphs \( G \) and \( G' \) that are neighboring, and for all \( O \in \text{Range}(\mathcal{A}) \),

\[
\Pr[\mathcal{A}(G) \in O] \leq e^{\epsilon} \cdot \Pr[\mathcal{A}(G') \in O].
\]

A differentially private algorithm \( \mathcal{A} \) provides privacy because, given any two graphs which differ on a single edge only, respective results of a same query on the graphs are not distinguishable. Therefore, an adversary cannot infer the value of any single edge in the dataset. Here, \( \epsilon \) represents the level of privacy. A smaller value of \( \epsilon \) means better privacy, but it also implies lower accuracy of the query result. The composition of differentially private algorithms also provides differential privacy, but it produces different results depending on the data to which the queries are applied.

- **Sequential composition** [12, Theorem 3]. Let \( \mathcal{A} \) each provides \( \epsilon_i \)-differential privacy. The sequence of \( \mathcal{A}_i(X) \) provides \( (\sum \epsilon_i) \)-differential privacy.

While there are many approaches to achieving differential privacy, the best known and most-widely used two for this purpose are the Laplace mechanism [13] and the exponential mechanism [14]. For real valued functions, i.e., \( f : \mathcal{G} \rightarrow R^d \), the most common way to satisfy differential privacy is to inject carefully chosen random noise into the output. The magnitude of the noise is adjusted according to the global sensitivity of the function, or the maximum extent to which any one tuple in the input can affect the output. Formally,

**Definition 2:** (Global Sensitivity [13]): The global sensitivity of the function \( f : \mathcal{G} \rightarrow R^d \) is \( \Delta(f) = \max_{G,G'} \|f(G) - f(G')\| \) for all neighboring \( G, G' \in \mathcal{G} \), where \( \| \cdot \| \) denotes the \( L_1 \) norm.

Similarly, the local sensitivity and local sensitivity at distance \( t \) of function \( f \) are defined as follows.

**Definition 3:** (Local Sensitivity [15]): The local sensitivity of the function \( f : \mathcal{G} \rightarrow R^d \) is \( LS(G, f, t) = \max_{G', t} \|f(G) - f(G')\| \) at distance \( t \) from \( G \) to \( G' \), where \( \| \cdot \| \) denotes the \( L_1 \) norm.

**Definition 4:** (Local Sensitivity at distance \( t \) [15]): The local sensitivity of \( f \) at distance \( t \) is the largest local sensitivity attained on graphs at distance at most \( t \) from \( G \). Formally, the global sensitivity of the function \( f : \mathcal{G} \rightarrow R^d \) at distance \( t \) is \( \Delta(f) = \max_{G,G', \|d(G,G')\| \leq t} \|f(G) - f(G')\| \) , where \( \| \cdot \| \) denotes the \( L_1 \) norm.

Note that global sensitivity can be understood as the maximum of local sensitivity over the input domain, i.e.,

\[
\Delta(f) = \max_{G} LS(G, f, t).
\]

To maintain differential privacy, the Laplace mechanism adds noise drawn from the Laplace distribution into the data to be published. The influence of any single edge on the outcome will be masked and hidden by the Laplace noise. Let \( \text{Lap}(\lambda) \) be a random value sampled from a Laplace distribution with mean zero and scale \( \lambda \). The Laplace Mechanism through which \( \epsilon \)-differential privacy is achieved is outlined in the following theorem.

...
Theorem 1: \[12\] Let \( f : \mathcal{G} \rightarrow R^d \). A mechanism \( M \) that adds independently generated noise from a zero-mean Laplace distribution with scale \( \Lambda = \Delta(f)/\epsilon \) to each of the \( d \) output values \( f(G) \), i.e., which produces \( O = f(G) + (\text{Lap}(\Delta(f)/\epsilon))^d \) satisfies \( \epsilon \)-differential privacy.

The exponential mechanism \[13\] is useful for sampling one of several options in a differentially-private way. A score to each of the options, which is determined by the input of the algorithm, is assigned by a quality function \( q \). Clearly, higher scores signify more desirable outcomes and the scores are then used to formulate a probability distribution over the outcomes in a way that ensures differential privacy.

Definition 5: (Exponential Mechanism \[13\]). Let \( q : (\mathcal{G} \times \mathcal{O}) \rightarrow R \) be a quality function that assigns a score to each outcome \( O \in \mathcal{O} \). Let \( \Delta_1(q) = \max_{d(G,G') \leq 1} \|q(G,O) - q(G',O)\| \) and \( M \) be a mechanism for choosing an outcome \( O \in \mathcal{O} \). Then the mechanism \( M \), defined by
\[
M(G,q) = \{ \text{return } O \text{ with probability } \propto \exp(q(G,O)/\epsilon) \}
\]
maintains \( \epsilon \)-differential privacy.

B. Related Work

With the increasing popularity of social network analysis research, privacy protection of social network data is a broad topic with a significant amount of prior work. In this section, we review the most closely related work about privacy protection on social network data.

An important thread of research aims to preserve social network data privacy by obfuscating the edges (vertices), i.e., by adding/deleting edges (vertices)\[6\], \[16\], \[17\], \[18\]. Mittal et al. proposed a perturbation method in \[16\] by deleting all edges in the original graph and replacing each edge with a fake edge that is sampled based on the structural properties of the graph. Liu et al. \[17\] design a system, called LinkMirage, which mediates privacy-preserving access to users social relationships in both static and dynamic social graphs. Hay et al. \[18\] perturb the graph by applying a sequence of \( r \) edge deletions and \( r \) edge insertions. The deleted edges are uniformly selected from the existing edges in the original graph while the added edges are uniformly selected from the non-existing edges. Wang et al. propose two different perturbation methods to anonymize social networks against CFP attacks in \[6\], which serves as the practical foundation for our algorithm. Their first method is based on adding dummy vertices, while the second algorithm achieves \( k \)-anonymity based on edge modification. Their proposed methods can resist CFP attacks on private users based on their connection information with the public users. Another important work for our algorithm is \[8\]. Yuan et al. \[8\] introduce a framework that provides personalized privacy protection for labeled social networks. They define three levels of privacy protection requirements by modeling gradually increasing adversarial background knowledge. The framework combines label generalization and other structure protection techniques (e.g., adding nodes or edges) in order to achieve improved utility.

Most of the obfuscating based works mainly focus on developing anonymization techniques for specific types of privacy attacks. They employ privacy models derived from \( k \)-anonymity \[19\] by assuming different types of adversarial knowledge. Unfortunately, all these anonymization techniques are vulnerable to attackers with stronger background knowledge than assumed, which has stimulated the use of differential privacy for more rigorous privacy guarantees.

There are many papers \[20\], \[21\], \[22\], \[23\], \[24\], \[25\] have started to apply differential privacy to protect edge/node privacy, defined as the privacy of users relationship or identity in graph data. One important application direction aims to release certain differentially private data mining results, such as degree distributions, subgraph counts and frequent graph patterns \[21\], \[23\], \[25\]. However, our problem is substantially more challenging than publishing certain network statistics or data mining results. Our goal is to publish the CFPs of private user, which incurs a much larger global sensitivity. Note that the sensitivity in the problem setting of \[23\] is only 1. Whats more, each private user in the networks independently specifies the privacy requirement for their data. In addition, some latest works are done for graph-oriented scenario. Proserpio et al. \[24\] develop a private data analysis platform wPINQ over weighted datasets, which can be used to answer subgraph-counting queries. Zhang et al. \[25\] propose a ladder framework to privately count the number of occurrences of subgraphs.

There are also some other related works aiming to publish a sanitized graph, which is out the scope of the objective of this paper. In \[20\], Sala et al. introduced Pygmalion, a differentially private graph model. Similar to \[26\], Wang and Wu employed the dK-graph generation model for enforcing edge differential privacy in graph anonymization \[27\]. In \[28\], Xiao et al. proposed a Hierarchical Random Graph (HRG) model based scheme to meet edge differential privacy. In addition, Chen et al. \[29\] propose a data-dependent solution by identifying and reconstructing the dense regions of a graph’s adjacency matrix.

III. PRIVACY GOAL

A. Problem Definition

In general, we assume there are some public users whose identities are not sensitive in social networks. Except the public users, the rest of the users are private users, whose edges are sensitive and each private user independently specifies the privacy requirement for their data. For convenience, we denote the set of public users in the social network as \( V_{\text{pub}} \) and the set of private users \( V_{\text{pri}} \) and we let \( |V_{\text{pub}}| = m_p \) and \( |V_{\text{pri}}| = m \). More formally, the Privacy Specification of private users is defined as follows:

Definition 6: (User-Level Privacy Specification\[9\]). A privacy specification is a mapping \( P : V_{\text{pri}} \rightarrow R^+ \) from private users to personal privacy preferences, where a smaller value represents a stronger privacy preference. The notation \( P^v \) is used to denote the privacy preference corresponding to user \( v \in V_{\text{pri}} \).
Similar to [9], we also describe a specific instance of a privacy specification as a set of ordered pairs, e.g., \( P := \{(v_1, \epsilon_1), (v_2, \epsilon_2), \ldots \} \) where \( v_i \in V_{pri} \) and \( \epsilon_i \in \mathbb{R}^+ \). We also assume that a privacy specification contains a privacy preference for every \( v \in V_{pri} \), or that a default privacy level is used. Here the information about any edge in \( G \) should be protected, and the privacy specification of edge \( e(v_i, v_j) \) can be quantified by \( \min\{P^v_i, P^v_j\} \).

In this paper, we focus on privately releasing of connection statistics between private users and public users. First, we specify the hop distance \( h(v_i, v_j) \) between two vertices \( v_i \) and \( v_j \) as the number of edges on the shortest path between them. Second, as indicated in ref. [6], we call the public user \( v \) as \( v_i \)'s connection fingerprint (CFP) if the private user \( v_i \) and the public user \( v \) is linked by a path of certain length. For a given hop distance \( n \), the formal definitions of the \( n \)-hop connection fingerprint \( CFP_n(v_i) \) and \( n \)-range connection fingerprint \( CFP(v_i, n) \) for a private user \( v_i \) in Definition 7 and Definition 8, respectively.

**Definition 7:** \((n\text{-th Hop Connection Fingerprint})\): the \( n \)-hop connection fingerprint \( CFP_n(v_i) \) of a private vertex \( v_i \) in a social network \( G = (V, E) \) consists of the group of public vertices whose hop distances to \( v_i \) are exactly \( n \), i.e., \( CFP_n(v_i) = \{ID(v_j) | v_j \in V_{pub} \land h(v_i, v_j) = n\} \).

**Definition 8:** \((n\text{-Range Connection Fingerprint})\): The \( n \)-range connection fingerprint of a private vertex \( v_i \), denoted by \( CFP(v_i, n) \), is formed by \( v_i \)'s \( x \)-th hop connection fingerprints, where \( 1 \leq x \leq n \), i.e., \( CFP(v_i, n) = \bigcup_{x \in [1, n]} CFP_x(v_i) \).

Given a system parameter \( c \), we aim to release the number of \( k \)-th hop \((1 \leq k \leq c)\) connection fingerprints for each private user in the sensitive graph, while protecting individual privacy in the meantime. Formally, we write \( f_k(G) : G \rightarrow R^m \) to denote the function that computes the number of \( k \)-th hop connection fingerprints for each private user in graph \( G \). Therefore, the final publication results for a sensitive graph \( G \) can be denoted in a form of \( m \times c \) matrix \( F = (f_1(G), \ldots, f_c(G)) \).

### B. Privacy Goal

The goal of this paper is to release the connection statistics under the novel notation of Personalized Differential Privacy (PDP) [9] introduced by Jørgensen et al. In contrast to traditional differential privacy, in which the privacy guarantee is controlled by a single, global privacy parameter (i.e., \( \epsilon \)), PDP makes use of a privacy specification, in which each user in \( V_{pri} \) independently specifies the privacy requirement for their data. More formally, the definition of PDP is showed in Definition 9.

**Definition 9:** \((Personalized Differential Privacy (PDP))\): In the context of a privacy specification \( P \) and a universe of private users \( U \), a randomized mechanism \( M : G \rightarrow R^m \) satisfies \( P \)-personalized differential privacy (or \( P \)-PDP), if for every pair of neighboring graphs \( G \) and \( G' \), with \( G \sim_{\epsilon_i} G' \) and \( e_{ij} = e(v_i, v_j) \), and for all \( O \in Range(M) \), \( Pr[M(G) \in O] \leq e^{\min\{P^v_i, P^v_j\}} \cdot Pr[M(G') \in O] \).

Intuitively, PDP offers the same strong, semantic notion of privacy that traditional differential privacy provides, but the privacy guarantee for PDP is personalized to the needs of every user simultaneously. Jørgensen et al. point out that the composition properties of traditional differential privacy extend naturally to PDP, see Theorem 2.

**Theorem 2:** \((Composition)\). Let \( M_1 : G \rightarrow R^m \) and \( M_2 : G \rightarrow R^m \) denote two mechanisms that satisfy PDP for \( P_1 \) and \( P_2 \), respectively. Then, the mechanism \( M_3 := (M_1(G), M_2(G)) \) satisfies \( P_3 \)-PDP, where \( P_3 = P_1 + P_2 \).

A smart general purpose mechanism for achieving PDP, called Sample Mechanism, is proposed in [9]. The sample mechanism works by introducing two independent sources of randomness into a computation: (1) non-uniform random sampling at the tuple level, and (2) additional uniform randomness introduced by invoking a traditional differentially private mechanism on the sampled input.

**Theorem 3:** \((The Sample Mechanism)\). Consider a function \( f : G \rightarrow R^m \), a social network \( G \in G \), a configurable threshold \( t \) and a privacy specification \( P \). Let \( RS(G, P, t) \) denote the procedure that independently samples each edge \( e_{ij} = e(v_i, v_j) \in G \) with probability \( \pi(e_{ij}, t) = \frac{e^{\min(P^v_i, P^v_j)}}{1 - e^{t - 1}} \) if \( \min\{P^v_i, P^v_j\} < t \), otherwise.

Given our problem of releasing the number of \( k \)-th hop \((1 \leq k \leq c)\) connection fingerprints under \( P \)-PDP, we overview the baseline and our advanced methods to give a brief glance on our motivations.

**Baseline method.** A baseline method is to apply an uniform or exponential budget allocation method and release a \( P/c \) (or \( P/2^k \)) and the budget is \( P/2^{c-1} \) indeed for \( k = c \)-personalized differential private result for every \( k \). If each released statistics for \( k \)-th hop \((1 \leq k \leq c)\) connection fingerprints preserves \( P/c \) (or \( P/2^k \))-PDP, the series of \( c \) counting queries guarantee \( P \)-PDP by theorem 1. These two baseline methods are denoted as Uniform and Exponential, correspondingly. Both Uniform and Exponential methods are easy to achieve \( P \)-PDP, but they ignore the fact that the statistics may not change significantly for successive queries \( f_k \) due to the sparsity of social network and inherits a large quantity of unnecessary noises.

**Distance-based method.** In this paper, we use a distance-based budget allocation approach inspired by [30] to reduce noise. Our proposed DEBA starts by distributing the publication budget in an exponentially decreasing fashion to every
private counting query $f_k(1 \leq k \leq c)$, i.e., query $f_k$ receives $P/2^{k+1}$ to publish its counting results. If it is found out that the distance between the statistics for $f_k$ and $f_{k-1}(2 \leq k \leq c-1)$ is smaller than its publication threshold, the counting query $f_k$ is skipped and its corresponding publication budget becomes available for a future counting query. On the other hand, if it is decided to publish the counting results of $f_k$, the $f_k$ should absorb all the budgets that became available from the previous skipped counting queries, and uses it in order to publish the current counting query $f_k$ with higher accuracy.

The presence or absence of one edge in the graph can contribute to a large number of potential CFPs, i.e., the global sensitivity of the counting queries is large and so the noise added to the count has to be scaled up. Our second method, DUBA-LF, further improves DEBA and uses a new technique, called ladder functions, for producing differentially private output. The technique specifies a probability distribution over possible outputs that are carefully defined to maximize the utility for the given input, while still providing the required privacy level. Moreover, we start the DUBA-LF by uniformly distributing the budget instead of the exponentially distributing in DEBA.

V. PROPOSED METHODS

We propose a distance-based budget absorption approach to release the number of $k$th-hop($1 \leq k \leq c$) CFPs under $\mathcal{P}$-PDP. Instead of releasing a $\mathcal{P}/c$ (or $\mathcal{P}/2^k$)-PDP result for every $k(1 \leq k \leq c)$, the new publication results are computed if and only if the distance between the counting statistics and the latest released statistics is larger than a threshold. It is worth noting that the statistics may not change significantly for successive queries $f_k$ due to the sparsity of social network. Therefore, this distance-based budget allocation approach can save some privacy budgets for a future counting query and reduce the overall error of released statistics.

In this section, our basic method, called DEBA, is presented at first. The basic method starts by exponentially distributing the budget to every private counting query $f_k(1 \leq k \leq c)$, and the budget absorption is decided by the distance between the counting statistics and the latest released statistics. We then introduce our advanced method, DUBA-LF, which uses ladder function to reduce the noise introduced by the traditional differentially private mechanism.

A. DEBA

DEBA (Publication with Distance-based Exponential Budget Absorption) starts with an exponentially decreasing budget for every private counting query $f_k(1 \leq k \leq c)$, and a privacy-preserving distance calculation mechanism is adopted to measure the distance between the counting statistics and the latest released statistics. The decision step uses the distance to decide whether to publish the private counting results of $f_k$ or not. If the decision is not, the private counting results of $f_k$ are approximated with the last non-null publication and the budget of $f_k$ becomes available for a future counting query. Otherwise, the private counting query $f_k$ absorbs all the budgets that are available from the previous skipped counting queries. The overall privacy budget is divided between the decision and publishing steps which are designed to guarantee personalized differential privacy as we will analyze later.

Before introducing the proposed DEBA, we give the sensitivity of counting query $f_k(1 \leq k \leq c)$ at first. Neighbor graphs of $G$ are all the graphs $G'$ which differ from $G$ by at most a single edge. For the counting query $f_1$, it queries the number of 1st-hop connection fingerprints for each private user in the sensitive graph, and changing a single edge in $G$ will result in at most one entry changing in the 1st-hop connection fingerprints. Hence, $\Delta(f_1) = 1$. For the counting query $f_k(2 \leq k \leq c)$, changing a single edge in $G$ will result in at most $|V_{pub}| = m_p$ entries changing in the $k$th-hop connection fingerprints, i.e., $\Delta(f_k) = m_p$ for $2 \leq k \leq c$.

Algorithm 1 presents the pseudocode of DEBA. DEBA is decomposed into two sub mechanisms: personal private distance calculation mechanism $M_1$ and personal private publication mechanism $M_2$. Line 1-5 capture the calculation of personal private distance between the counting statistics and the latest released statistics, labeled as mechanism $M_1$. Line 6-9 carry out the publication step for 1st-hop connection fingerprints and line 10-21 carry out the publication step for $k$th-hop($2 \leq k \leq c$) connection fingerprints. Line 11 or line 17 gets the total budgets of skipped queries whose budgets for publication is absorbed. Then the publication threshold $T_k$ for query $f_k$ is determined by $m_p/\epsilon_k$. The reason to define such a threshold is that the injecting Laplace noise of $f_k$ is with scale $T_k$. Then DEBA compares the distance $dist$ to the threshold $T_k$ (line 12). If the distance is larger than $T_k$, DEBA samples the private social network $G$ (Line 13) and outputs the noisy counts (Line 14), or null otherwise (Line 20). In addition, DEBA outputs $t$-th-hop connection fingerprints with the totally remaining budgets as shown in Line 16-19.

Remark: Recall that the error of randomly sampling input graph $G$ is data-dependent as well as the error of distance based approximation. And we cannot present a formal utility analysis for such a data-dependent mechanism. We will present extensive experiments using real datasets to justify the performance of our algorithms. Moreover, precisely optimizing $\ell$ for an arbitrary $f$ may be nontrivial in practice because, although the error of $DP^\ell_f$ may be quantified without knowledge of the dataset, the impact of sampling does depend on the input data. A possible option, in some cases, is to make use of old data that is no longer sensitive (or not as sensitive), and that comes from a similar distribution, to approximately optimize the threshold without violating privacy. It is demonstrated that for many functions, the simple heuristics of setting $t = \max_t, P^w$ or $t = \frac{1}{m} \sum_{w} P^w$, often give good results on real data and privacy specifications[6].

B. DUBA-LF

The proposed DEBA mechanism publishes the private count of $f_k$ by adding Laplace noise to the true answer, where the scale of noise is proportional to the global sensitivity of $f_k$. It is pointed out that the global sensitivity of $f_k$ is 1 for
Algorithm 1 Pseudocode of DEBA

Input:
- $G$ - private input graph
- $V_{pri,G}$ - the set of private users in $G$
- $V_{pub,G}$ - the set of public users in $G$
- $P$ - the privacy specification
- $m$ - the cardinality of $V_{pri,G}$
- $c$ - the counting range
- $t$ - the configurable sample threshold

Output: $\hat{F} = (\hat{F}_1, \hat{F}_2, \ldots, \hat{F}_c)$

For each $k (1 \leq k \leq c)$ do

//personal private distance calculation mechanism $M_1$
1: Calculate $F_k = f_k(G)$.
2: Identify last non-null release $\tilde{F}_j$ from $\tilde{F}$.
3: Sample $c_k = RS(F_k, \mathcal{P}/2c, t/2c)$ with probability $\pi(v_i, t/2c)$.
4: Set $dist = \frac{1}{2} \sum_{j=1}^{m} |\tilde{F}_j[j] - c_k[j]|$ and $\epsilon_{k,1} = t/2c$.
5: Calculate $dist = dist + \text{Lap}(2m_p/c/mt)$.

//personal private publication mechanism $M_2$
6: if $k = 1$ then
7: Set $\epsilon_{k,2} = t/4$.
8: Sample $G_k = RS(G, \mathcal{P}/4, t/4)$ with probability $\pi(e_{ij}, t/4)$.
9: Calculate $F_k = \tilde{f}_k(G) = f_k(G_k) + \text{Lap}(\frac{1}{t})$.
10: else if $k < c$ then
11: Calculate $\epsilon_{k,2} = \sum_{j=r+1}^{k} \frac{t}{2^{j-r}}$ and $T_k = m_p/\epsilon_{k,2}$.
12: if $dist > T_k$ then
13: Sample $G_k = RS(G, \mathcal{P}, \sum_{j=r+1}^{k} \frac{t}{2^{j-r}}, \epsilon_{k,2})$ with probability $\pi(e_{ij}, \epsilon_{k,2})$.
14: Calculate $F_k = f_k(G) = f_k(G_k) + \text{Lap}(\frac{m_p}{\epsilon_{k,2}})$.
15: end if
16: else if $k = c$ then
17: Calculate $\epsilon_{k,2} = \sum_{j=r+1}^{k} \frac{t}{2^{j-r}}$ and $T_k = m_p/\epsilon_{k,2}$.
18: Sample $G_k = RS(G, \mathcal{P}, \sum_{j=r+1}^{k} \frac{t}{2^{j-r}}, t/4)$ with probability $\pi(e_{ij}, \epsilon_{k,2})$.
19: Calculate $F_k = f_k(G) = f_k(G_k) + \text{Lap}(\frac{m_p}{\epsilon_{k,2}})$.
20: else set $\hat{F}_k = null$.
21: end if

$k = 1$ or $m_p$ for $2 \leq k \leq c$. It is obvious that there can be numerous public users in large network graphs. Hence, the global sensitivity of counting query $f_k$ may be very large and makes the noise large enough to overwhelm the true answer. In order to improve the utility of private release for $f_k$, we use the new definition of ladder function to reduce the introduced noise. The definition of ladder function is presented at first.

Definition 10: (Ladder function). A function $I_x(G)$ is said to be a ladder function of query $f$ if and only if

(a) $LS(G, f) \leq I_0(G)$, for any $G$;
(b) $I_x(G) \leq I_{x+1}(G)$, for any pair of neighboring graphs $G, G'$, and any nonnegative integer $x$.

A straightforward example of a ladder function for count query $f_k$ is $I_1(G, f_k) = \Delta(f_k)$, since $LS(G, f) \leq \Delta(f_k)$ for any $G$, and a constant always satisfies the second requirement. However, as aforementioned, the global sensitivity of counting query $f_k$ can be extremely large for CFP counting, which may not require so much noise.

For counting query $f_1$, its global sensitivity is 1 and the ladder function for $f_1$ can be defined as $I_1(G, f_1) = 1$. Before detailing the ladder function for $f_k (2 \leq k \leq c)$, the important notion local sensitivity is refined by defining the sensitivity for a particular pair of nodes by defining the sensitivity for a particular pair of nodes $(v_i, v_j)$, denoted by $LS_{ij}(G, f)$. Then $LS(G, f) = \max_{i,j} LS_{ij}(G, f)$. Let $p_i$ denotes the number of the number of 1st-hop connection fingerprints for user $v_i$ and $d_{max}$ be the maximum node degree in $G$. Then it is easy to get that $LS(G, f_k) = \max_i p_i$ for $f_k (2 \leq k \leq c)$. Without losing of generality, we simply assume that $p_i \leq p_j$ for a particular pair of nodes $(v_i, v_j)$. Then we give our ladder function for $k$-th-hop connection fingerprint counting queries $f_k (2 \leq k \leq c)$ in Theorem 4 and prove that the constructed ladder function satisfy the requirements in Definition 10.

Theorem 4: $I_1(G, f_k) = \min\{m_p, LS(G, f_k) + 1\}$ is a ladder function for $f_k (2 \leq k \leq c)$.

Proof: The proof contains the following two steps.

(i) $LS(G, f_k) \leq I_0(G, f_k)$ for any $G$. This step is trivial since $I_0(G, f_k) = LS(G, f_k)$.
(ii) $I_x(G', f_k) \leq I_{x+1}(G, f_k)$ for any neighboring graphs $G$ and $G'$. Note the fact that the set $\{G^*, d(G^*, G') \leq t\}$ is a subset of $\{G^*, d(G^*, G) \leq t + 1\}$. Therefore, $\max\{\sum_{i=0}^{x} I_x(G, f_k) - f_k(G)\} \leq \max\{d(G, G') \leq t + 1, I_{x+1}(G, f_k)\}$, i.e., $I_1(G, f_k) = \min\{m_p, LS(G, f_k) + t + 1\} = \min\{m_p, LS(G, f_k) + t + 1\}$.

It is clear that the ladder function $I_1(G, f_k)$ converges to $\Delta(f_k)$ when $t \geq \Delta(f_k) - LS(G, f_k)$. The ladder function $I_1(G, f_k)$ is used to determine the quality function $q$ in exponential mechanism and define how $q$ varies. In particular, $q$ is a symmetric function over the entire integer domain, centered at $f_k(G)$. The quality function $q$ is defined as follows:

Definition 11: (Ladder Quality). Formally, given ladder function $I_1(G, f_k)$ we define the ladder quality function $q_{f_k}(G, v_i, s)$ for node $v_i$ by

(i) $q_{f_k}(G, v_i, f_k(v_i)) = 0$;
(ii) for $s \in f_k(v_i) \pm \{\sum_{t=0}^{x} I_x(G, f_k)\}$, set $q_{f_k}(G, v_i, s) = -u - 1$.

After assigning each integer a quality score, the sensitivity of the quality function can be calculated as, $\Delta(q_{f_k}) = \max_{v_i, G, G'} |q_{f_k}(G, v_i, s) - q_{f_k}(G', v_i, s)| = 1$. We refer the reader to [25] (THEOREM 4.2) for a full description of the proof of it.

DUBA-LF (Publication with Distance-based Uniformly Budget Absorption using Ladder Function) uses ladder function to reduce the introduced noise while reallocating the pre-allocated uniform privacy budget. The pseudocode of DUBA-LF is presented in Algorithm 2. The personal private distance calculation mechanism $M_1$ is identical to that of DEBA (Lines 1-5 in Algorithm 1). The personal private publication mechanism $M_2$ is presented in Lines 6-21. Lines 6-9 carry out
Algorithm 2: Pseudocode of DUBA-LF

**Input:**
- $G$ — private input graph
- $V_{pri,G}$ — the set of private users in $G$
- $V_{pub,G}$ — the set of public users in $G$
- $\mathcal{P}$ — the privacy specification
- $m$ — the cardinality of $V_{pri,G}$
- $c$ — the counting range
- $t$ — the configurable sample threshold

**Output:** $\hat{F} = (\hat{F}_1, \hat{F}_2, \ldots, \hat{F}_c)$

For each $k(1 \leq k \leq c)$ do:
- \text{// personal private distance calculation mechanism $M_1$
- \text{// Same as Lines 1-5 in Algorithm 1
- \text{// personal private publication mechanism $M_2$

6: if $k = 1$ then
7: Set $\epsilon_{k,2} = t/2c$.
8: Sample $G_k = RS(G, \mathcal{P}/2c, t/2c)$ with probability \[ \pi(\epsilon_{ij}, t/2c). \]
9: Calculate $\hat{F}_k = \bar{f}_k(G) = f_k(G_k) + \text{Lap}(\frac{\epsilon_k}{t}).$
10: end if
11: else if $k < c$ then
12: Calculate $\epsilon_{k,2} = (k-r)t/2c$ and $T_k = m_p/\epsilon_{k,2}$.
13: if $d > T_k$ then
14: Sample $G_k = RS(G, \mathcal{P} \cdot (k-r)/c, \epsilon_{k,2})$ with probability \[ \pi(\epsilon_{ij}, \epsilon_{k,2}). \]
15: Calculate $\hat{F}_k = \text{LFNoising}(f_k(G_k), \epsilon_{k,2}, I_x(G, f_k)).$
16: end if
17: else if $k = c$ then
18: Sample $G_k = RS(G, \mathcal{P} \cdot (k-r)/c, \epsilon_{k,2})$ with probability \[ \pi(\epsilon_{ij}, \epsilon_{k,2}). \]
19: Calculate $\hat{F}_k = \text{LFNoising}(f_k(G_k), \epsilon_{k,2}, I_x(G, f_k)).$
20: end if

end for

Algorithm 3: $\text{LFNoising}(f_k(G_k), \epsilon_{k,2}, I_x(G, f_k))$

**Input:**
- $f_k(G_k)$ — the number of $k$th-hop connection fingerprints in sampled input graph $G_k$
- $\epsilon_{k,2}$ — the notional publication budget
- $I_x(G, f_k)$ — the ladder function

**Output:** $\tilde{F}_k = (\tilde{F}_k[1], \tilde{F}_k[2], \ldots, \tilde{F}_k[m])$

1: Set $d = 0$, range$[0] = f_k(G_k)$ and weight$[0] = \exp(\frac{\epsilon_k}{2c} \cdot 0)$.
2: for $x = 1$ to $M = m_p - \text{LS}(G, f_k)$ do
3: range$[x] = f_k(G_k) \pm (d, d + I_x(G, f_k)^m)$.
4: weight$[x] = 2I_{x-1}(G, f_k) \exp(\frac{\epsilon_k}{2} \cdot (-x))$.
5: $d = d + I_x(G, f_k)$.
6: end for
7: weight$[M + 1] = \frac{2m_p \exp(\frac{\epsilon_k}{2} \cdot (-M-1))}{1 - \exp(-\frac{\epsilon_k}{2})}$.
8: Randomly sample $m$ integers $\tilde{T} = (t_1, t_2, \ldots, t_m)$. Here $t_i(1 \leq i \leq m)$ is drawn with probability weight$[t_i]$ over sum of weights.
9: for $i = 1$ to $m$ do
10: if $t_i \leq M$ then uniformly sample an integer $j$ from range$[t_i][i]$ and set $\tilde{F}_k[i] = j$;
11: else sample an integer $h$ from the geometric distribution with parameter $p = 1 - \exp(-\frac{\epsilon_k}{2})$.
12: uniformly sample an integer $j$ from $f_k(G_k)[i] \pm d + hm_p + (0, m_p)$ and set $\tilde{F}_k[i] = j$.
13: end if
14: end for
15: return $\tilde{F}_k$.

The publication step for counting query $f_1$. The publication step for $k$th-hop ($2 \leq k \leq c$) connection fingerprints is carried out in Line 10-21. DUBA-LF samples the private social network $G$ (Line 13 or Line 18) in the same way with DEBA but the sampling probabilities are different. The personal private publication for $k$th-hop ($2 \leq k \leq c$) connection fingerprints in DUBA-LF is also different to DEBA. If the distance is larger than $T_k$ or counting for $f_c$, DUBA-LF uses an exponential mechanism based mechanism $\text{LFNoising}$ to provide differential privacy (Line 14 and Line 19). In the meantime, DUBA-LF outputs null if the distance is not larger than $T_k$ (Line 20).

$\text{LFNoising}$ is an extending algorithm of $\text{NoiseSample}$ in [25]. $\text{NoiseSample}$ is proposed to output one value as the final differentially private result while our proposed $\text{LFNoising}$ is aims to solve the problem of differentially private releasing in the vector form. The pseudocode of $\text{LFNoising}$ is presented in Algorithm 3. Given the ladder function $I_x(G, f_k)$, the calculation of the range and weight for the first few rungs, e.g., rung 0 (the center) to rung $M + 1$ ($M = m_p - \text{LS}(G, f_k)$), are shown in Lines 1-7. Lines 8-12 describe the random sampling of the private publication vector $\tilde{F}_k$ which presents the private count of $k$th-hop ($2 \leq k \leq c$) connection fingerprints.

VI. PRIVACY ANALYSIS

The proofs of privacy guarantees for the proposed mechanisms are formally provided in this section. We show the proposed DEBA satisfies P-personalized differential privacy first.

Lemma 1: Mechanism $M_1$ in Algorithm 1 is $\mathcal{P}/2$-personalized differentially private.

Proof: We use the notation $F_{k,-v}$ and $F_{k,+v}$ to mean the graph resulting from removing from or adding to $F_k$ the tuple $f_k(v)$. We can represent two neighboring datasets (vectors) as $F_k$ and $F_{k,-v}$. For each $1 \leq k \leq c$, all of the possible outputs of $RS(F_k, \mathcal{P}/2c, t/2c)$ can be divided into those in which $f_k(v)$ was selected, and those in which $f_k(v)$ was not selected. The sensitivity of dist function in $M_1$ is $m_p/m$, therefore, the mechanism injects Laplace noise with scale $2m_p/c/mt$ in Line 5 can be denoted as $D^P_{\text{dist}}$. Thus, we have...
Differential privacy is captured in Lemma 1. Mechanism every personalized differential privacy. To prove that DEBA satisfies \( \tilde{\Delta} \) differentially private if it publishes, \( 0 - \) personalized is trivially equal to zero, as no publication occurs. In the latter case, the privacy budget is trivially equal to zero, as no publication occurs. Theorem 5:

\[
\Pr[S_{F_k}(F_k, P/2c, t/2c) \in O]
= \Pr[DP_{dist}^{2/2c}(RS(F_k, P/2c, t/2c)) \in O]
= \sum_{Z \in F_k \rightarrow v} (\pi(v, t/2c) \Pr[RS(F_k, P/2c, t/2c) = Z]).
\]

There are two cases for \( v \) that we must consider:

1. \( P^v/2c \geq t/2c \),
2. \( P^v/2c < t/2c \).

For the former case, we have \( \pi(v, t/2c) = 1 \) and eq. (1) can be rewritten as

\[
\Pr[S_{F_k}(F_k, P/2c, t/2c) \in O] \\
\leq (1 - e^{P^v/2c}) \Pr[S_{F_k}(F_k, P/2c, t/2c)] \\
= e^{P^v/2c} S_{F_k}(F_k, P/2c, t/2c) \\
\leq e^{P^v/2c} S_{F_k}(F_k, P/2c, t/2c) \\
\leq e^{P^v/2c} S_{F_k}(F_k, P/2c, t/2c)
\]

For the latter case \( P^v/2c < t/2c \),

\[
\Pr[S_{F_k}(F_k, P/2c, t/2c) \in O] \\
\leq \Pr[DP_{dist}^{2/2c}(Z+v) \in O] + \sum_{Z \in F_k \rightarrow v} ((1 - \pi(v, t/2c)).
\]

There are two cases for \( v \) that we must consider:

1. \( \Pr[P_{dist}^{2/2c}(Z+v) \in O] \)
2. \( \Pr[P_{dist}^{2/2c}(Z+v) \in O] \)

Lemma 2: LFNoising satisfies \( \epsilon_{k,2} - \) differentially private.

\[
\text{Proof: There are two steps in the algorithm LFNoising: selecting a rung of the ladder (where rung \( M+1 \) is considered as a special case) according to the relative value of the weight of the rung and picking an integer from the corresponding rung. For rungs 0 to M, the possible output values on the same rungs are picked uniformly. For rung \( M+1 \), the possible outputs are determined by two actions picking how many further rungs down the ladder to go and then picking uniformly from these. As discussed above, for \( 1 \leq i \leq m \), the output probability distribution is equal to}
\]

\[
\Pr[F_k[i] = \rho] = \frac{\exp \left( \frac{s_{k+2}}{2 \Delta(q_{k+2})} \cdot q_{k+2}(G, v, \rho) \right)}{\sum_{v} \exp \left( \frac{s_{k+2}}{2 \Delta(q_{k+2})} \cdot q_{k+2}(G, v, \rho) \right)}
\]

As argued above, if the input graph \( G \) is replaced by its neighboring graph \( G \), the quality of \( \rho \) will be changed by at most \( \Delta (q_{k+2}) = 1 \), i.e., the numerator \( \exp \left( \frac{s_{k+2}}{2 \Delta(q_{k+2})} \cdot q_{k+2}(G, v, \rho) \right) \) can change at most \( \exp \left( \frac{s_{k+2}}{2 \Delta(q_{k+2})} \cdot \Delta(q_{k+2}) \right) = \exp \left( \frac{s_{k+2}}{2} \right) \).

Moreover, a single change in graph \( G \) the changing in denominator is minimized by a factor of \( \exp \left( -\frac{s_{k+2}}{2} \right) \), giving the ratio of the new probability of \( \rho \) and the original one \( \exp(\ell_{k,2}) \).

Therefore, LFNoising \( (f_k(G_k), \epsilon_{k,2}, I_1(G, f_k)) \) is \( \epsilon_{k,2} - \) differentially private.

This result highlights the fact that LFNoising \( (f_k(G_k), \epsilon_{k,2}, I_1(G, f_k)) \) can be denoted as \( D_{P_{dist}}^{k+2} \).

Similar to Theorem 5, we can conclude that DUBA-LF is \( \mathcal{P} \)-personalized differentially private.

Theorem 6: The proposed DUBA-LF satisfies \( \mathcal{P} \)-PDP.

\[ \text{Proof: The proof is similar to that of Theorem 5 and we omit it.} \]

VII. EXPERIMENTAL EVALUATION

We make use of three real-world graph datasets in our experiments: polblogs[31], facebook[32] and CondMat[33].
networks. The **polblogs** network was crawled from the US political blogosphere in 2005. The vertices are blogs of a set of US politicians, and an edge between two blogs represents the existence of hyperlinks from one blog to the other. The **facebook** network was collected from the survey participants using a Facebook app. The vertices are Facebook users, and an edge between two users represents the established friendship between them. The **CondMat** network was collaboration networks from the e-print arXiv, which cover scientific collaborations between authors who submitted papers to Condensed Matter category. The edge between two authors represents an author co-authored a paper with another author in this network. All the networks are represented by undirected and unweighted graphs with no isolated vertices.

The real-world networks we used do not contain public user identity. In other words, all vertices in the networks are anonymous. In order to evaluate the proposed CFPs publication algorithms, we select a set of vertices in each network assuming their identities are public. Thereafter, based on these public vertices, we generate the CFPs of the remaining private vertices. In this paper, we select vertices with the highest degrees as public vertices and the proportion of public users is set to 5% in our whole experiment. Table I presents some basic statistics of the networks.

| Dataset | nodes | edges | diameter | density | \(|V_{pub}\) |
|---------|-------|-------|----------|---------|-------------|
| Polblogs| 1222  | 16724 | 8        | 2.24%   | 61          |
| Facebook| 4039  | 88234 | 8        | 1.08%   | 201         |
| CondMat | 23133 | 93439 | 14       | 0.0349% | 1156        |

We compared DEBA and DUBA-LF with benchmarks **Uniform** and **Exponential** over these three datasets. We implemented all methods in Matlab, ran each experiment 100 times, and reported the average error, expressed as the Mean Absolute Error (MAE) and the Mean Relative Error (MRE). To generate the privacy specifications for our experiments, we randomly divided the private users (records) into three groups: conservative, representing users with high privacy concern; moderate, representing users with medium concern; and liberal, representing users with low concern. The fraction of each type users is 1/3. The privacy preferences for the users in the conservative, moderate and liberal groups received a privacy preference of \(\epsilon_c = 1\), \(\epsilon_m = 4\) and \(\epsilon_l = 16\) respectively. As a result, the average privacy preference of all users equals to 7.

![Fig. 1. MAE and MRE of each mechanism for the Polblogs dataset, as sample threshold is varied and \(c\) is set to 4.](image1)

![Fig. 2. MAE and MRE of each mechanism for the Polblogs dataset, as sample threshold is varied and \(c\) is set to 7.](image2)

![Fig. 3. MAE and MRE of all schemes for the Polblogs dataset, where we vary the sampling threshold \(t\) and set \(c = 4\). DEBA seems to outperform the other methods in this setting. And DEBA outperforms Uniform mechanism by up to 45.8% in MAE and 126.3% in MRE, Exponential by up to 42.9% in MAE and 125.0% in MRE.](image3)

Fig. 1 plots the MAE and MRE of all schemes for the Polblogs dataset, where we vary the sampling threshold \(t\) and set \(c = 4\). DEBA seems to outperform the other methods in this setting. And DEBA outperforms Uniform mechanism by up to 45.8% in MAE and 126.3% in MRE, Exponential by up to 42.9% in MAE and 125.0% in MRE, and DUBA-LF by up to 25.4% in MAE and 76.7% in MRE. Similar to Fig.2, increasing the sampling threshold has a less evident effect for both Uniform and Exponential mechanisms. We can also conclude that increasing the sampling thresholds causes decreasing of MAR and MRE for DUBA-LF and DEBA for small threshold \(t\) while increasing of MAR and MRE for larger \(t\).
The MAE and MRE of all schemes for the Facebook dataset is showed in Fig.4, and here we set $c = 4$ and vary the sampling threshold $t$. DUBA-LF seems to outperform the other methods in this setting. It outperforms Uniform mechanism by up to 27.2% in MAE and 86.3% in MRE, Exponential by up to 17.7% in MAE and 70.0% in MRE, and DEBA by up to 4.1% in MAE and 3.9% in MRE. Similar to Fig.1, increasing the sampling thresholds causes decreasing of MAR and MRE for DUBA-LF and DEBA. However, increasing the sample threshold has an evident effect for decreasing of MAR and MRE for both Uniform and Exponential for small threshold $t$ while increasing of MAR and MRE for larger $t$.

Fig.6 shows the MAE and MRE of all schemes for the CondMat dataset, where the sample threshold $t$ is varied and $c$ is set to 7. DUBA-LF is shown to be the best method in this setting. And it outperforms Uniform mechanism by up to 120.5% in MAE and 894.5% in MRE, Exponential by up to 76.9% in MAE and 862.7% in MRE, and DEBA by up to 34.4% in MAE and 84.3% in MRE. It is indicated that increasing the sampling threshold has a little effect of decreasing of MAR and MRE for Uniform and Exponential mechanisms. We can also conclude that increasing the sampling thresholds causes decreasing of MAR and MRE for DUBA-LF and DEBA for small threshold $t$ while increasing of MAR and MRE for larger $t$.

VIII. Conclusion

The number of CFPs is one of the most important properties for a public users labeled graph. In order to release the number of CFPs in the context of personalized privacy preferences, we proposed two schemes (DEBA and DUBA-LF) to achieve personal differential privacy. Both DEBA and DUBA-LF use the distance-based budget absorption mechanism to improve the publication utility while DUBA-LF also employs ladder function to reduce the introduced noise. We formally prove that the proposed DEBA and DUBA-LF schemes are $P$-PDP and we conduct thorough experimentation with real datasets, which demonstrated the superiority and the practicality of our proposed schemes.

REFERENCES

[1] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou R3579X? . In WWW, 2007.
[2] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Wei. Resisting structural re-identification in anonymized social networks. In VLDB, 2008.
[3] G. Cormode, D. Srivastava, S. Bhagat, and B. Krishnamurthy. Class-based graph anonymization for social network data. In VLDB, 2009.
[4] L. Zou, L. Chen, and M. Özsu. K-automorphism: A general framework for privacy preserving network publication. In VLDB Endowment, vol. 2, no. 1, pp. 946-957, 2009.
[5] J. Cheng, A. W.-c. Fu, and J. Liu, K-isomorphism: privacy preserving network publication against structural attacks. In SIGMOD10, 2010, pp. 459-470.
[6] Wang Y, Zheng B. Preserving privacy in social networks against connection fingerprint attacks (C)/Data Engineering (ICDE), 2015 IEEE 31st International Conference on. IEEE, 2015: 54-65.
[7] C. Dwork. Differential privacy. In ICALP, pages 1-12, 2006.
[8] Yuan M, Chen L, Yu P S. Personalized privacy protection in social networks[J]. Proceedings of the VLDB Endowment, 2010, 4(2): 141-150.
[9] Jorgensen Z, Yu T, Cormode G. Conservative or liberal? Personalized differential privacy[C]/Data Engineering (ICDE), 2015 IEEE 31st International Conference on. IEEE, 2015: 1023-1034.
[10] G. Times, Media, govt, organizations get hooked on weibo: report. 2013. [Online]. http://www.globaltimes.cn/content/757560.shtml
[11] H. Bunke. On a relation between graph edit distance and maximum common subgraph. Pattern Recogn. Lett., 18(9):689C694, Aug. 1997.
[12] F. D. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In Proc. of ACM SIGMOD, Jun.29-Jul.2009, Providence, Rhode Island
[13] C. Dwork, et al. Calibrating noise to sensitivity in private data analysis. In Proceedings of TCC. Springer, 2006, pp. 265-284.
[14] F. McSherry and K. Talwar. Mechanism design via differential privacy, in FOCS, 2007, pp. 94-103
[15] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In STOC, pages 75C84, 2007.
[16] Mittal P, Papamanthou C, Song D. Preserving Link Privacy in Social Network Based Systems[J]. Computer Science - Cryptography and Security, 2012.
[17] Liu C, Mittal P. LinkMirage: Enabling Privacy-preserving Analytics on Social Relationships[C]. NDSIS.2016.
[18] Hay M, Miklau G, Jensen D, et al. Anonymizing social networks[J]. Technical Report, University of Massachusetts, Amherst, 2007.
[19] L. Sweeney, K-anonymity: A Model for Protecting Privacy, IJUFKS, vol. 10, no. 5, pp. 557C570, 2002.
[20] M. Hardt and A. Roth. Beating randomized response on incoherent matrices. In STOC, 2012.
[21] M. Hay, C. Li, G. Miklau, and D. Jensen. Accurate estimation of the degree distribution of private networks. In ICDM, 2009.
[22] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith. Analyzing graphs with node differential privacy. In TCC, 2013.
[23] E. Shen and T. Yu. Mining frequent graph patterns with differential privacy. In SIGKDD, 2013.
[24] D. Proserpio, S. Goldberg, and F. McSherry. Calibrating data to sensitivity in private data analysis: A platform for differentially-private analysis of weighted datasets. In VLDB, 2014.
[25] J. Zhang, G. Cormode, C. Procopiuc, D. Srivastava, and X. Xiao. Private release of graph statistics using ladder functions. In SIGMOD, 2015.
[26] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Y. Zhao. Sharing graphs using differentially private graph models. In IMC, 2011.
[27] Y. Wang and X. Wu. Preserving differential privacy in degree-correlation based graph generation. TDP.2013.
[28] Q. Xiao, R. Chen, and K. Tan. Differentially private network data release via structural inference. KDD,2014.
[29] Chen R, Fung B C M, Yu P S, et al. Correlated network data publication via differential privacy[J]. Vldb Journal, 2014, 23(4):653-676.
[30] G. Kellaris, et al. Differentially private event sequences over infinite streams. PVldb,7(12): 1155-1166(2014)
[31] Adamic L. A, Glance N. The political blogosphere and the 2004 US election: divided they blog[C].Proceedings of the 3rd international workshop on Link discovery. ACM, 2005: 36-43.
[32] McAuley J J, Leskovec J. Learning to Discover Social Circles in Ego Networks[C].NIPS. 2012, 2012: 548-56.
[33] Leskovec J, Kleinberg J, Faloutsos C. Graph evolution: Densification and shrinking diameters[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2007, 1(1): 2.