Applications of Differential Privacy in Social Network Analysis: A Survey

Honglu Jiang, Jian Pei, *Fellow, IEEE,* Dongxiao Yu,*Member, IEEE,*
Jiguo Yu, *Senior Member, IEEE,* Bei Gong, and Xiuzhen Cheng, *Fellow, IEEE*

Abstract—Differential privacy provides strong privacy preservation guarantee in information sharing. As social network analysis has been enjoying many applications, it opens a new arena for applications of differential privacy. This article presents a comprehensive survey connecting the basic principles of differential privacy and applications in social network analysis. We concisely review the foundations of differential privacy and the major variants. Then, we discuss how differential privacy is applied to social network analysis, including privacy attacks in social networks, models of differential privacy in social network analysis, and a series of popular tasks, such as analyzing degree distribution, counting subgraphs and assigning weights to edges. We also discuss a series of challenges for future work.

Index Terms—Differential privacy; social network data analysis; global sensitivity; smooth sensitivity; local differential privacy; dependent differential privacy; degree distributions; subgraph counting; edge weight query.

1 Introduction

As a reflection of real social life, social networking provides a vehicle to share a lot of private and sensitive information [1]. For example, in many online social networking sites, a user is required to provide personal information such as name, gender, birthdate, education level, marital status, personal photo, or even cell phone number. Besides, user-generated contents such as texts, pictures, videos, and geographical locations are also retained in the databases [2]. Such data is often shared with third parties for additional business services, such as data analysis, targeted advertising, recommendations and evaluations on apps. If personal private information is leaked or abused, involved individuals may become victims of intrusion attacks, such as spamming mails, junk messages and telephone harassments. In some extreme cases, damages to personal reputation, properties, or even physical injuries may be caused due to illegal data disclosures [3].

The problem of data privacy protection was first put forward by Dalenius [4] in the late 1970s, who pointed out that the purpose of protecting private information in a database is to prevent any user, including legitimate users and potential attackers, from obtaining accurate information about arbitrary individuals when accessing the database. Following this principle, many privacy preservation models with strong operability were proposed, such as k-anonymity [5], l-diversity [6], t-closeness [7] and (α, k)-anonymity [8]. However, each of those models provides protection against only a specific type of attacks and cannot defend against newly developed ones. A fundamental cause of this deficiency lies in that the security of a privacy preservation model relies on an assumption of some specific background knowledge of an attacker. Nevertheless, it is almost impossible to enumerate all possible types of background knowledge that an attacker may have. Therefore, a privacy preservation model independent of background knowledge is highly desirable.

Dwork developed *differential privacy* [18] to provide a strong privacy guarantee and protect against the privacy disclosure of statistical databases. Under differential privacy, query results of a dataset are insensitive to the change of a single record. That is, whether a single record exists in the dataset has little effect on the output distribution of the analytical results. An attacker cannot obtain accurate individual information by observing the results because the risk of privacy disclosure caused by adding or deleting a single record is kept within a user-specified, acceptable range. Differential privacy assumes that an attacker can obtain all information in a dataset except for the target record, which can be regarded as the maximum background knowledge that an attacker may have. It rests on a sound mathematical foundation under certain assumptions as well as quantitative evaluations. Differential privacy is a standard for quantifying privacy risks rather than a single tool and has been widely used in statistical estimations, data publishing, data mining and machine learning. There exist many methods and implementations to achieve differentially private data analysis.

Differential privacy mainly aims at statistical problems in databases at first. Because of its unique strengths, dif-

H. Jiang is with the Department of Computer Science, The George Washington University, DC, USA, 20052, USA. E-mail: hljiang0720@gwu.edu
J. Pei is with the School of Computing Science, Simon Fraser University, Burnaby, B.C. Canada V5A 1S6. Email: jpei@cs.sfu.ca
D. Yu (Corresponding Author) is with the School of Computer Science & Technology, Shandong University, Qingdao, 266237, P.R. China. E-mail: dxyu@sdu.edu.cn
J. Yu (Corresponding Author) is with the Qilu University of Technology (Shandong Academy of Sciences), Jinan, Shandong, 250353, P.R. China; and with Shandong Laboratory of Computer Networks, Jinan, 250014, P. R. China. Email: jiguoyu@sina.com
B. Gong is with the School of Information Technology, Beijing University of Technology, Beijing 100124, P.R. China. Email: gongbei@bjut.edu.cn
X. Cheng is with the Department of Computer Science, The George Washington University, DC, USA, 20052, USA; and with the School of Computer Science and Technology, Shandong University, Qingdao, 266510, P.R. China. E-mail: cheng@gwu.edu, xzcheng@sdu.edu.cn
Manuscript received ; revised .

TABLE 1
Summary on Previous Survey Articles

Topic	Ref	Focuses	Major angles
Differential Privacy Techniques	[9]	Basic techniques to achieve differential privacy and applications in statistical datasets.	Learning theory, Statistical datasets
	[10]	Differential privacy theory and application on two aspects of statistical datasets, privacy preserving data release and privacy preserving data mining.	Statistical data publishing and mining, Applications of differential privacy
	[11]	A comprehensive survey on differential privacy techniques for cyber-physical systems.	Application and implementation in cyber-physical systems
Privacy of Social Networks	[12–15]	Surveys on privacy risks and anonymization techniques for privacy preserving publishing of social network data.	Privacy risks, Anonymization in social networks
	[16]	A survey on graph data anonymization, de-anonymization attacks and de-anonymizability quantification.	Anonymization, De-anonymization
	[17]	A review on the privacy risks that exist in different aspects of social media data such as attribute and identity disclosure attacks.	Privacy risks, Social media

ferential privacy has been applied to social network data analysis. A number of suitable adaptations of differentially private social network analysis techniques have been developed [19–25]. Nevertheless, social networks post a series of challenges for privacy preserving analysis. Generally speaking, social networks can be modeled as graphs and become very complicated at large scale; they often have strong data correlations since social relationships among users are not independent. Particularly, as demonstrated by Liu *et al.* [26], the dependence among tuples in statistical databases may seriously weaken the privacy guarantee that current differential privacy mechanisms provide. This obviously also holds for social networks.

Therefore, there exist at least three fundamental challenges that should be tackled in order to apply differential privacy to social network data analysis. First, we have to adapt differential privacy from tabular data to network data. Second, we have to address the issue of high sensitivity in complex and correlated social network data. Last, we have to explore the tradeoff between data utility and privacy guarantee as too much noise added for differential privacy guarantee may make query results useless.

Comprehensively understanding differential privacy and its applications in social network data analysis is far from trivial. There exist multiple relevant surveys on differential privacy [9–11] and privacy preservation in social network analysis [12–17], whose topics, focuses and major angles are summarized in Table 1. One can see that these existing surveys focus on either differential privacy in tabular statistical databases or privacy preservation on social network analysis. More specifically, most surveys on differential privacy focus on its theory, basic techniques and applications on statistical datasets, such as differential privacy for various queries; those on privacy preservation in social networks mainly focus on privacy risks and attacks, de-anonymization and graph data anonymization techniques in social networks. Although differential privacy in social network analysis has become an active and influential area, to the best of our knowledge, no existing survey is fully dedicated to applications of differential privacy in social network analysis.

This motivates our endeavor in this article, whose major objective is to provide intuitive interpretations and illustrations on the important ideas in differential privacy, especially noise calibration to global sensitivity and smooth sensitivity, and review the state-of-the-art applications of differentially privacy in social network analysis addressing the three major challenges mentioned above. In our survey, we explore the interplay between differential privacy and social network analysis by systematically introducing the theoretical basis of differential privacy and comprehensively reviewing differentially private methods on social network analysis. Specifically, we summarize four models of differential privacy definitions from the perspective of social graphs including node privacy, edge privacy, outlink privacy and partition privacy; then carefully divide the existing differentially private methods for three most widely-used graph analysis techniques (degree distribution, counting subgraphs and social network weights) into different categories based on their technical strategies such as post-processing, bounded-degree graph and random matrix projection. Moreover, we summarize the research results of local differential privacy on social network analysis, which has gained significant attentions in recent years as a promising approach for privacy-preserving data publishing.

Conducting research on differential privacy in social networks needs real social network data. The Stanford Network Analysis Platform (SNAP)[1] provides an extensive repository [27]. It includes a few popular online social networks, communication networks, citation networks, web and blog datasets, and several other large network datasets.

The rest of the article is organized as follows. In Section 2, we review the basic concepts of differential privacy with detailed interpretations and examples. More specifically, we define and explain the differential privacy model, describe its noise mechanisms calibrated to global sensitivity and smooth bounds of local Sensitivity, and present the composition properties. For better elaboration, we exemplify popular functions such as count and median, and detail the corresponding differentially private noise mechanisms.

In Section 3, we discuss two most popular variants of differential privacy. *Dependent differential privacy* is proposed to handle queries involving correlated database tuples. *Local differential privacy* is a well-developed extension to centralized differential privacy.

In Section 4, to effectively demonstrate how to adapt differential privacy from tabular data to social network data, we first summarize the popular privacy attacks in

1. http://snap.stanford.edu/data/

TABLE 2
An Example Database

Name	Disease or Not
Ross	1
Monica	1
Bob	0
Joey	0
Alice	1

social networks, and then introduce the four models of network privacy, namely *node privacy*, *edge privacy*, *out-link privacy*, and *partition privacy*. We illustrate the definitions of these graph differential privacy models and analyze their applicability and complexity.

In Section 5, we provide an overview on differentially private algorithms for *analyzing degree distribution*, *counting subgraphs* and *assigning weights to edges*, the three most widely-used graph analysis techniques under the social network privacy models mentioned in Section 4. Our analysis demonstrates that most of them may not be able to obtain good utility due to large network size, complex graph structures and strong graph attribute correlations.

In Section 6, we summarize the state-of-the-art research results in local differential privacy and point out the challenges of applying local differential privacy to social network analysis.

In Section 7, we conclude this article and present several open research challenges.

2 Differential Privacy

In this section, we review the core concepts in differential privacy. We exemplify popular query functions, such as count and median, to illustrate the corresponding noise mechanisms calibrated to *global sensitivity* and *smooth sensitivity*.

2.1 Intuition

An individual's information may be inferred even without explicitly querying for the specific details. For example, consider the data in Table 2, which is about whether a person suffers from a disease. Suppose the database provides a query interface $Q_i(D)$, which returns the sum of the second column, 'Disease or Not', of the first i rows. The query returns an aggregate and does not explicitly query about any specific person.

Suppose an attacker somehow knows the background knowledge that the record about Alice is the last row in the database, and wants to infer whether or not Alice has the disease. The attacker can issue two queries $Q_5(D)$ and $Q_4(D)$, and compute $Q_5(D) - Q_4(D)$. Alice has the disease if the outcome is 1 and she does not have the disease otherwise. This simple example shows how personal information may be disclosed even when it is not explicitly queried. It is not safe to release exact query answers even when data is not published.

The intuition of differential privacy is to inject a controlled level of statistical noise into query results to hide the consequence of adding or removing an arbitrary individual from a dataset. That is, when querying two almost identical datasets (differing by only one record, for example), the results are differentially privatized so that an attacker cannot glean any new knowledge about an individual with high probability, i.e., whether or not a given individual is present in the dataset cannot be guessed with useful confidence. In the example shown in Table 2, to protect Alice's privacy, one can inject noises into answers to $Q_5(D)$ and $Q_4(D)$ so that $Q_5(D) - Q_4(D)$ and Alice's value on the column 'Disease or Not' are independent with high probability.

2.2 Definition of Differential Privacy

Let f be a query function to be evaluated on a dataset D. We want to have an algorithm A running on the dataset D and returning $A(D)$ such that $A(D)$ should be $f(D)$ with a controlled amount of random noise added. The goal of differential privacy is to make $A(D)$ close to $f(D)$ as much as possible to ensure data utility, and at the same time $A(D)$ should preserve the privacy of the entities in the dataset.

Differential privacy mainly addresses adversarial attacks that queries datasets differing by only a small number of entries. There are two flavors of differential privacy, namely *unbounded* and *bounded*, which are distinguished by the definition of *neighboring datasets* [28]. For two datasets D and D', if D' can be obtained by adding or removing a tuple from D, it is called unbounded. If D' can be obtained by changing the value of a tuple from D, then it is called bounded. That is, bounded neighboring datasets have the same size while the sizes of two unbounded neighboring datasets differ by 1. There exist slight differences in presenting the query results for unbounded and bounded neighboring datasets, but the ideas of designing and analyzing the differential privacy mechanisms are the same. Therefore in this article, we employ both types of neighboring datasets to illustrate the introduced differential privacy mechanisms.

Definition 1 (Differential privacy [29]). A randomized algorithm A is *ϵ-differentially private* if, for any two neighboring datasets D and D' and any subset S of possible outputs of A,

$$Pr[A(D) \in S] \leq e^{\epsilon} Pr[A(D') \in S],$$

where $\epsilon \geq 0$ is a parameter called *privacy budget*.

Privacy budget ϵ in Definition 1 is often a small positive real number that reflects the level of privacy preservation algorithm A can provide. For example, if $\epsilon = 0.01$, $e^{0.01} \approx 1.01$; and 0.01-differential privacy ensures that the distributions of $A(D)$ and $A(D')$ are very similar and almost indistinguishable. The smaller the value of ϵ, the higher the level of privacy preservation. A smaller ϵ provides greater privacy preservation at the cost of lower data accuracy since more noise has to be added. When $\epsilon = 0$, the level of privacy preservation reaches the maximum, that is, "perfect" protection. In this case, the algorithm outputs two results with indistinguishable distributions but the corresponding results do not reflect any useful information about the dataset. Therefore, the setting of ϵ should balance the tradeoff between privacy and data utility. In practical applications, ϵ usually takes small values such as 0.01, 0.1, or $\ln 2$, $\ln 3$ [9]. Computing ϵ-differential privacy may be

challenging in some scenarios. To facilitate approximation, a generalized notion of differential privacy is developed.

Definition 2 (Approximate differential privacy [30]). A randomized algorithm A is (ϵ, δ)-differentially private if, for any two neighboring datasets D and D' and any subset S of possible outputs of A,

$$Pr[A(D) \in S] \leq e^{\epsilon} Pr[A(D') \in S] + \delta$$

When $\delta > 0$, (ϵ, δ)-differential privacy relaxes ϵ-differential privacy by a small probability controlled by parameter δ. In ϵ-differential privacy, the ratio between the output probability distributions for neighboring datasets D and D' is strictly bounded by e^{ϵ}; while in (ϵ, δ)-differential privacy, a freedom to breach the strict ϵ-differential privacy for certain low probability events is offered. That is, in (ϵ, δ)-differential privacy, equation $Pr[A(D) \in S] \leq e^{\epsilon} Pr[A(D') \in S]$ holds with the probability at least $1 - \delta$.

Typically, δ is set to far smaller than the inverse of any polynomial in the size n of the database (i.e., $\delta \ll \frac{1}{p(n)}$) [31]. An equivalent formulation states that δ is cryptographically negligible when $\delta \leq n^{-\omega(1)}$ [31]. Note that $\frac{1}{p(n)}$ can be described as an upper bound of δ since a value of δ in the order of $\frac{1}{p(n)}$ is dangerous for privacy leakage.

Differential privacy can be achieved by adding an appropriate amount of noise to query results, that is, $A(D) = f(D) + Z$. Adding too much noise may hurt data utility, while adding too little noise cannot provide sufficient privacy guarantee. Sensitivity, which captures the largest change to the query results caused by adding/deleting or changing any record in the dataset, is the key parameter to determine the magnitude of the added noise. Accordingly, global sensitivity, local sensitivity, and smooth sensitivity are defined under differential privacy.

2.3 Noise Calibration

How to add noise to query results $f(D)$ and how much noise should be added are the key to the noise mechanisms in differential privacy. In this subsection, we introduce two frameworks of differentially private noise mechanisms, namely, noise calibration to global sensitivity and noise calibration to smooth sensitivity.

2.3.1 Noise Calibration to Global Sensitivity

Definition 3 (Global sensitivity [18]). For $f : D \rightarrow R^d$, the global sensitivity of f for all pairs of neighboring datasets D and D' is

$$GS_f = \max_{D,D'} \|f(D) - f(D')\|_1,$$

where $\| \cdot \|_1$ denotes the L_1 norm.

The global sensitivity measures the maximum change of query results when modifying one tuple. It is related only to the query function, and is independent from the dataset. For some functions such as sum, count, and max, the global sensitivity is easy to compute. For instance, the global sensitivity for counting is 1 since only one tuple is changed for any two neighboring datasets, and that for the histogram query is 2, where the change is measured by the L_1 norm between two histogram vectors. For some other functions such as maximum diameter of k-means clusters and subgraph counting, the global sensitivity may be difficult to compute or be unbounded. For example, the median function can have a high global sensitivity. Take $f(D) = median(x_1, x_2, \ldots, x_n)$ as an example, of which x_i is a real number in $[0, M]$. Assume that n is an odd number and that $x_1, x_2, \ldots, x_n$ are sorted. Thus, $f(D) = x_m$, where $m = \frac{n+1}{2}$. Consider the following extreme case,

$$\begin{aligned} D &: \{0, 0, \ldots, x_m = 0, x_{m+1} = M, \ldots, M\}, \\ D' &: \{0, 0, \ldots, x_{m-1} = 0, x_m = M, \ldots, M\}. \end{aligned}$$

We have $f(D) = 0$ and $f(D') = M$. Therefore, the global sensitivity for this function is M, which can be arbitrarily large in general. As another example, the global sensitivity of the triangle counting query of a graph is unbounded, since the change of triangle counts depends on the graph size.

The noise injected to achieve differential privacy can be calibrated according to the global sensitivity of the query function, that is, the maximum amount of change to the query result when only one record is modified in the dataset. For a function with a small global sensitivity, only a small amount of noise needs to be added to cover up the impact on query results when one record is changed. However, when the global sensitivity is large, it is necessary to add a substantial amount of noise to the output to ensure the privacy guarantee, which leads to poor data utility. Two noise mechanisms, namely Laplace mechanism [29] and exponential mechanism [32], were respectively proposed for numerical and categorical query results.

2.3.1.1 Laplace Mechanism: The Laplace distribution [33] (centered at μ) with scale b is the distribution with probability density function

$$h(z) = \frac{1}{2b} \exp(-\frac{|z - \mu|}{b}).$$

Denote by $Lap(b)$ the Laplace distribution (centered at 0) with scale b. Dwork *et al.* [29] proposed the Laplace mechanism, which states that for dataset D and function $f : D \rightarrow R^d$ with global sensitivity GS_f, $A(D) = f(D) + Z$ is ϵ-differentially private, where $Z \sim Lap(GS_f/\epsilon)$.

The Laplace mechanism is suitable for protecting numerical results. Considering the counting function as an example, since the global sensitivity of counting is $GS_f = 1$, if we choose $\epsilon = 0.1$, the Laplace mechanism outputs $3 + Lap(10)$ for the specific D.

2.3.1.2 Exponential Mechanism: In some situations, query results are categorical, such as finding the zip code of the highest average income. McSherry *et al.* [32] developed the exponential mechanism for the situations where the "best" needs to be selected. Let $Range$ be the output domain of a query function and each value $r \in Range$ be an entity object. In the exponential mechanism, the *utility function* of the output value r, denoted by $q(D, r)$, is employed to evaluate the quality of r. Given a randomized algorithm A with input dataset D and output entity object $r \in Range$, let Δq be the global sensitivity of function $q(D, r)$. McSherry *et al.* [32] showed that, if an algorithm A selects and outputs r from $Range$ at a probability proportional to $\exp(\frac{\epsilon q(D,r)}{2\Delta q})$, then A is ϵ-differentially private.

Table 3 presents an example of the exponential mechanism. Consider a basket $\mathcal{D}$ with three kinds of fruits: apple ($\mathcal{A}$), banana ($\mathcal{B}$), and cherry ($\mathcal{C}$). Algorithm A seeks to output the kind of fruits that has the largest amount. Let $q(D, \mathcal{A}) = count(\mathcal{A})$, $q(D, \mathcal{B}) = count(\mathcal{B})$, and $q(D, \mathcal{C}) = count(\mathcal{C})$.

TABLE 3
An Example of Exponential Mechanism

Item	$q(D,r)$ ($\Delta q=1$)	Probability		
		$\epsilon=0$	$\epsilon=0.1$	$\epsilon=1$
$\mathcal{A}$	10	$exp(\frac{0\times10}{2\times1})=1$, $Pr[output=\mathcal{A}]=\frac{1}{1+1+1}=1/3$	$exp(\frac{0.1\times10}{2\times1})=e^{0.5}$, $Pr[output=\mathcal{A}]=\frac{e^{0.5}}{e^{0.5}+e^{1}+e^{1.5}}=0.186$	$exp(\frac{1\times10}{2\times1})=e^{5}$, $Pr[output=\mathcal{A}]=\frac{e^{5}}{e^{5}+e^{10}+e^{15}}=4.509\times10^{-5}$
$\mathcal{B}$	20	$exp(\frac{0\times20}{2\times1})=1$, $Pr[output=\mathcal{B}]=\frac{1}{1+1+1}=1/3$	$exp(\frac{0.1\times20}{2\times1})=e^{1}$, $Pr[output=\mathcal{B}]=\frac{e^{1}}{e^{0.5}+e^{1}+e^{1.5}}=0.307$	$exp(\frac{1\times20}{2\times1})=e^{10}$, $Pr[output=\mathcal{B}]=\frac{e^{10}}{e^{5}+e^{10}+e^{15}}=0.0686$
$\mathcal{C}$	30	$exp(\frac{0\times30}{2\times1})=1$, $Pr[output=\mathcal{C}]=\frac{1}{1+1+1}=1/3$	$exp(\frac{0.1\times30}{2\times1})=e^{1.5}$, $Pr[output=\mathcal{C}]=\frac{e^{1.5}}{e^{0.5}+e^{1}+e^{1.5}}=0.506$	$exp(\frac{1\times30}{2\times1})=e^{15}$, $Pr[output=\mathcal{C}]=\frac{e^{15}}{e^{5}+e^{10}+e^{15}}=0.993$

Thus $\Delta q=1$, since adding or removing an apple, a banana or a cherry causes a change of the utility function value to be at most 1. Based on the exponential mechanism, one can compute the probabilities of outputting $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ with a given ϵ, which are shown in Table 3.

The output probability of the item with a high utility function is amplified when ϵ is large, such as when $\epsilon=1$ in the table. As ϵ decreases, the utility differences of the items become more and more smoothed and the probabilities of the outputs tend to be equal. When $\epsilon=0$, the output probabilities for all items are equal.

2.3.2 Noise Calibration to Smooth Sensitivity

When the global sensitivity is large, a substantial amount of noise has to be added to the output to achieve differential privacy, which may seriously impair data utility. To address this issue, Nissim *et al.* [22] proposed the idea of local sensitivity, the sensitivity with respect to a given data set.

Definition 4 (Local sensitivity [22]). The local sensitivity of function $f: D\rightarrow R^d$ on D is

$$LS_f(D)=\max_{\text{neighboring data set } D' \text{ of } D}\|f(D)-f(D')\|_1$$

Let us take the median function as an example, that is, $f(D)=median(x_1,x_2,\ldots,x_n)$, where n is odd and $x_1,x_2,\ldots,x_n$ are sorted. We have $f(D)=x_m$, where $m=\frac{n+1}{2}$, and $LS_f(D)=\max\{x_m-x_{m-1},x_{m+1}-x_m\}$.

The local sensitivity is related to not only the query function f but also the given dataset D. According to Definition 3, $GS_f=\max_D(LS_f(D))$. Since the magnitude of noise is proportional to sensitivity, the amount of noise added is much less with local sensitivity. Unfortunately, local sensitivity does not satisfy the requirement of differential privacy, because the noise magnitude itself may reveal the database information. For example, consider a database where the values are between 0 and $M>0$, and two neighboring databases $D(0,0,0,0,0,M,M)$ and $D'(0,0,0,0,M,M,M)$. Let f be the median function. Then, $f(D)=0$ and $f(D')=0$, and the corresponding local sensitivities are respectively $LS_f(D)=0$ and $LS_f(D')=M$. Thus if the noises are calibrated according to 0 and M, respectively, to compute $A(D)$ and $A(D')$, then they are easy to be distinguished by an adversary. An algorithm A is not (ϵ,δ)-differentially private if local sensitivity is adopted.

To bridge the gap, a *smooth upper bound* of the local sensitivity is proposed to determine the magnitude of the added noise [22].

Definition 5 (Smooth bound and smooth sensitivity [22]). For a dataset D and a query function f, a function $S: D\rightarrow R$ is a β-smooth upper bound of $LS_f(D)$ with $\beta>0$, if $\forall D, S(D)\geq LS_f(D)$ and for all bounded neighboring datasets D, D', $S(D)\leq e^{\beta}S(D')$.

The β-smooth sensitivity of function f with $\beta>0$ is

$$S^*_{f,\beta}(D)=\max_{D'}\{LS_f(D')\cdot e^{-\beta\cdot d(D,D')}\}.$$

When $\beta=0$, $S(D)$ becomes the constant GS_f to satisfy the requirements in Definition 5. Obviously, global sensitivity is a simple but possibly loose upper bound on LS_f. When $\beta>0$, the smooth sensitivity is a conservative upper bound on LS_f. LS_f may have multiple smooth bounds, and the smooth sensitivity is the smallest one that meets Definition 5.

Again, consider the median function as an example. We construct a function $A^{(k)}(D)$ that calculates how much the sensitivity can change when up to k entries are modified.

$$A^{(k)}(D)=\max_{D'\in\mathbb{D}:d(D,D')\leq k}LS_f(D')$$

where $\mathbb{D}$ is the domain of all possible datasets. Then, the smooth sensitivity can be expressed using $A^{(k)}(D)$ as

$$S^*_{f,\beta}(D)=\max_{k=0,\ldots,n}e^{-k\beta}A^{(k)}(D).$$

To compute $A^{(k)}(D)$, we need to calculate the maximum of $LS_f(D')$ where D' and D differ by up to k tuples. Recall that D is sorted, $f(D)=x_m$, and $LS_f(D)=\max\{x_m-x_{m-1},x_{m+1}-x_m\}$. Thus, we have

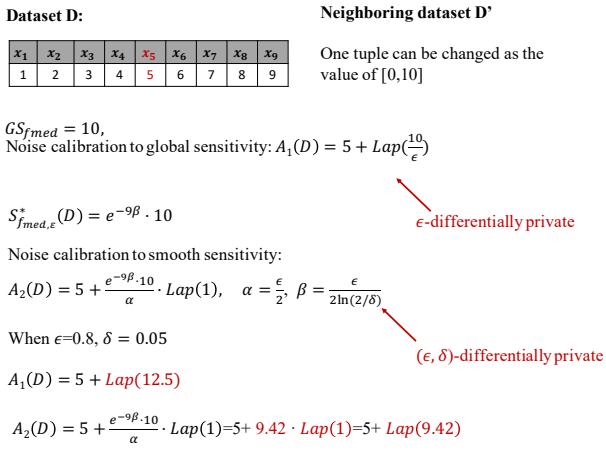
$$A^{(k)}(D)=\max_{0\leq t\leq k+1}\{x_{m+t}-x_{m+t-k-1}\}.$$

Then, the smooth sensitivity of the median function can be calculated by

$$S^*_{f_{med},\beta}(D)=\max_{0\leq k\leq n}(e^{-k\beta}\cdot\max_{0\leq t\leq k+1}(x_{m+t}-x_{m+t-k-1}))$$

In general, computing smooth sensitivities for functions such as counting the number of triangles in a graph is non-trivial and even NP-hard [34]. Therefore, a smooth upper bound is used to replace the smooth sensitivity when the latter is hard to compute. Next, we show how to employ a β-smooth sensitivity (or upper bound) to calibrate noise for ϵ-differential privacy.

According to the framework of differential privacy presented in Section 2.3.1, $A(D)=f(D)+Z$ is returned for query f on dataset D, where Z is a random variable drawn



x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
1	2	3	4	5	6	7	8	9

Fig. 1. Comparison of noise calibrations.

from a distribution. If $Z \sim Lap(GS_f/\epsilon)$, $A(D)$ provides ϵ-differential privacy. In ϵ-differential privacy, the magnitude of the added noise should be as small as possible to preserve data utility and should be independent of the database for strong privacy protection. Noise calibrated according to global sensitivity is independent from the database D but the magnitude may be too big making the query results unusable. Noise calibrated according to local sensitivity is dependent on D thus failing to provide differential privacy. To address this challenge, Nissim *et al.* [22] proposed to use noise calibrated according to the smooth upper bound of local sensitivity (more preferably, the smooth sensitivity). The basic idea is to add noise proportional to $\frac{S_f(D)}{\alpha}$, that is, $A(D) = f(D) + \frac{S_f(D)}{\alpha} \cdot Z$, where S_f is a β-smooth upper bound on the local sensitivity of f, Z is a random variable with probability density function h. Nissim *et al.* [22] pointed out that h must be (α, β)-*admissible* in order to achieve differential privacy based on smooth sensitivity.

Definition 6 ***((α, β)-admissible noise distribution [22]).*** For all $\Delta \in \mathcal{R}$ and $\lambda \in \mathcal{R}$ such that $|\Delta| \leq \alpha$ and $|\lambda| \leq \beta$, a probability density function h is (α, β)-admissible if it satisfies the following conditions:

$$\text{Sliding Property:} \quad h(z) \leq e^{\frac{\epsilon}{2}} \cdot h(z + \Delta) + \frac{\delta}{2}$$
$$\text{Dilation Property:} \quad h(z) \leq e^{\frac{\epsilon}{2}} \cdot (e^{\lambda} h(e^{\lambda} \cdot z)) + \frac{\delta}{2}$$

The sliding and dilation properties ensure that the noise distribution cannot change much under sliding and dilation, and the values of α and β are the upper bounds of Δ (the sliding offset) and λ (the dilation offset) based on h. If h of Z is (α, β)-admissible,the database access mechanism $A(D) = f(D) + \frac{S_f(D)}{\alpha} \cdot Z$ is (ϵ, δ)-differentially private [22].

There are three families of admissible distributions: Cauchy, Laplace, and Gaussian [22]. A Cauchy admissible distribution yields a "pure" ϵ-differential privacy with $\delta = 0$. Laplace and Gaussian admissible distributions can produce an approximate differential privacy with $\delta > 0$ under different α and β values.

Consider the median function again and let $D = (x_1, x_2, \ldots, x_n)$, where $x_1 \leq x_2 \leq \cdots \leq x_n$ and $x_i \in [0, M]$. The global sensitivity of the median function is M. Figure 1 illustrates the two differentially private mechanisms based on global sensitivity and smooth sensitivity, both utilizing

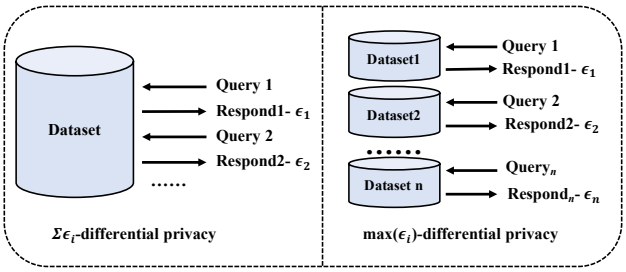


Fig. 2. Composition properties of differential privacy.

Laplace distributions. The noise calibrated to the smooth sensitivity is less, since the probability of the random variable Z taking a value closer to 0 is larger, and that of Z taking a larger value is smaller. Thus, more noise is added to the output of the global sensitivity based mechanism. In conclusion, for median, at the same privacy protection level (same ϵ), noise calibrated to smooth sensitivity has a smaller magnitude, thus better preserving data utility.

2.4 Composite Differential Privacy

Sometimes a complex privacy preservation problem needs a composite algorithm that involves more than one differential privacy algorithms. More specifically, one may need to sequentially apply various differential privacy algorithms to a dataset, or may need to run various differential privacy algorithms over disjoint datasets to solve a composite problem. The privacy budgets of the composite algorithms for these two cases are summarized by the following two theorems, and the basic concepts are further demonstrated in Figure 2.

Let $A_1, A_2, \ldots, A_n$ be n ϵ-differential privacy algorithms, whose privacy budgets are respectively denoted by $\epsilon_1, \epsilon_2, \ldots, \epsilon_n$.

Theorem 1. (Sequential Composition [35]) The composite algorithm obtained by sequentially applying $A_1, A_2, \ldots, A_n$ on the same dataset D provides $\sum_{i=1}^{n} \epsilon_i$-differential privacy.

Theorem 2. (Parallel Composition [35]) Let $D_1, D_2, \ldots, D_n$ be n arbitrary disjoint datasets. The composite algorithm obtained by applying each A_i on a corresponding D_i provides $\max\{\epsilon_i\}$-differential privacy.

The above two theorems provide the so-called "sequential composition" and "parallel composition" properties. Theorem 1 states the sequential composition property: the level of privacy preservation provided by a composite algorithm consisting of a sequence of differential privacy algorithms over the same dataset is determined by the sum of the individual privacy budgets. Theorem 2 presents the "parallel composition" property: when differential privacy algorithms are applied to disjoint datasets, the overall level of privacy preservation provided by the composite algorithm depends on the worst privacy guarantee among all the differential privacy algorithms, that is, the one with the largest privacy budget. These two theorems can be used to determine whether a composite algorithm satisfies the differential privacy requirement and to reasonably control the allocation of the total privacy budget to each algorithm.

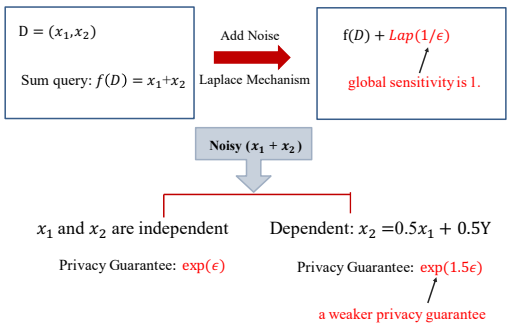


Fig. 3. Privacy guarantee for a dependent dataset.

3 Two Variations of Differential Privacy

To adapt to various problem domains and settings, different variations of differential privacy have been developed. In this section, we introduce two most popular variants: *dependent differential privacy* tends to handle queries involving correlated database tuples while *local differential privacy* targets the scenarios where an untrustworthy third-party is employed to collect data.

3.1 Dependent Differential Privacy

Differential privacy assumes that the tuples in a database are independent from each other. This assumption is not always true in practice. As indicated by Kifer and Machanavajjhala [28], the correlation or dependence between tuples may undermine the privacy guarantees of differential privacy mechanisms. Yang *et al.* [36] investigated the influence of data correlations on privacy and presented the notion of *Bayesian differential privacy*. They further proposed a Gaussian correlation model to accurately describe data correlations and developed a perturbation algorithm satisfying Bayesian differential privacy. Consider the following simple example [26, 36]. Let $D = (x_1, x_2)$ be a database, and tuples x_1 and x_2 have a probabilistic dependence $x_2 = 0.5x_1 + 0.5Y$, where x_1 and Y have uniform and independent distributions over $[0, 1]$, and Y is a random variable to model the relationship between x_1 and x_2 and to keep x_1 and x_2 in $[0, 1]$. The global sensitivity is 1. Consider the situation where the Laplace mechanism is applied to the sum query $f(D) = x_1 + x_2$. One can see from Figure 3 that the privacy guarantee is $\exp(1.5\epsilon)$ when x_1 and x_2 are considered correlated while it is $\exp(\epsilon)$ if we assume that x_1 and x_2 are independent.

Liu *et al.* [26] further developed the notion of *Dependent Differential Privacy* and the corresponding mechanisms. In a database D, if any tuple is dependent on at most $L - 1$ other tuples, the dependence size is L. Denote by R the *probabilistic dependence relationship* over the L dependent tuples. Two datasets $D(L, R)$ and $D'(L, R)$ are *dependently neighboring* if changing one tuple in $D(L, R)$ can impact at most $L - 1$ other tuples in $D'(L, R)$.

Definition 7 (Dependent Differential Privacy [26]). A randomized algorithm A is ϵ-dependent differentially private if for any two dependent neighboring datasets $D(L, R)$ and $D'(L, R)$, and for all sets S of possible outputs, we have

$$\max_{D(L,R),D'(L,R)} \frac{P(A(D(L,R)) = S)}{P(A(D'(L,R)) = S)} \leq \exp(\epsilon)$$

Dependent differential privacy limits the capacity of an adversary to infer sensitive information, and can defend against all possible adversarial inferences even if the adversary has full knowledge of the tuple correlations.

A Laplace mechanism achieving ϵ-dependent differential privacy for a dataset $D(L, R)$ with dependence size L and probabilistic dependence relationship R was proposed in [26]. Specifically, for a dataset $D(L, R)$ and a query function f with global sensitivity GS_f, an ϵ/L-differentially private Laplace mechanism $A(D) = f(D) + Lap(L \cdot GS_f/\epsilon)$ can achieve ϵ-dependent differential privacy.

Consider the example shown in Figure 3. The global sensitivity for the sum query is 1 and the dependence size $L = 2$. Thus, the output for this Laplace mechanism is $A(D) = f(D) + Lap(2/\epsilon)$. However, this mechanism implies that all the dependent tuples are *completely dependent* on each other, which makes the query sensitivity $L \cdot GS_f = 2$, while the sensitivity of the sum query for the two dependent tuples is 1.5. In real world datasets, there may be very few tuples that are completely dependent on each other, though they may be related. Thus, many mechanisms consider a fine-grained dependence relationship between tuples to obtain a small dependent sensitivity of queries. For example, Zhao *et al.* [37] adopted the probability graphical model to represent the dependency structure of tuples and achieved high utility.

3.2 Local Differential Privacy

The basic differential privacy setup relies on a trusted third party to collect data, add carefully crafted noise to a query result according to the specification of differential privacy, and publish the noisy statistical results. Nevertheless, in practice it is often difficult to find a truly trusted third party to collect and process data. The lack of trusted third parties greatly limits the applications of the basic, centralized differential privacy. To address this issue, *local differential privacy* [38] emerges, which does not assume the existence of any trusted third-party data collector. Instead, it transfers the process of data privacy protection to individual users by asking each of them to independently deal with and protect personal sensitive information. Figure 4 shows the framework of local differential privacy. One can see that local differential privacy extends its centralized counterpart by localizing perturbed data to resist privacy attacks from untrusted third-party data collectors.

3.2.1 The Definition of Local Differential Privacy

Definition 8 (Local Differential Privacy [38]). Consider n users, with each possessing one record. A randomized algorithm A with input and output domains $Dom(A)$ and $Ran(A)$, respectively, is said to satisfy ϵ-local differential privacy if the probability of A obtaining the same output result t^* ($t^* \subseteq Ran(A)$) on any two records t and t' ($t, t' \in Dom(A)$) satisfies

$$Pr[A(t) = t^*] \leq e^{\epsilon} \times Pr[A(t') = t^*]$$

Local differential privacy ensures the similarity between the output results of any two records. By this way it is

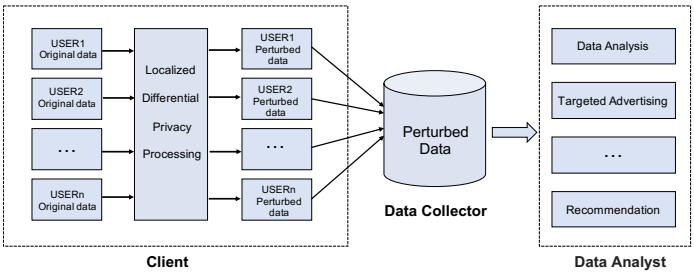


Fig. 4. A framework of local differential privacy.

almost impossible to infer which record is the input data according to an output result of algorithm A. In centralized differential privacy, the privacy guarantee of algorithm A is defined on neighboring datasets, and requires a trusted third-party data collector. Nevertheless, in local differential privacy, each user processes its individual data independently, that is, the privacy preserving process is transferred from the data collector to individual users, so that a trusted third party is no longer needed and privacy attacks brought from an untrusted third-party data collector is thus avoided. The implementation of local differential privacy requires data perturbation mechanisms.

3.2.2 Perturbation Mechanisms

The random response technique [39] proposed by Warner in 1965 is the mainstream perturbation mechanism adopted by local differential privacy. The main idea is to protect data privacy by making use of the uncertainty in the responses to sensitive questions. Consider an example of n persons with an unknown proportion π of diseased patients. To calculate π, a survey question is launched: "are you a patient with some disease?" Each user responds with either "Yes" or "No". For privacy preservation, a user may not respond with the true answer. Assume that a user responds with the help of a non-uniform coin flip in which the probability of heads showing up is p and the probability of tails showing up is $1 - p$. Then if a head shows up, the user responds with the true answer; otherwise, it responds with the opposite. The data collector aggregates all responses from the users and estimates the count of diseased persons. This mechanism achieves ϵ-local differential privacy, where $\epsilon = |\ln \frac{p}{1-p}|$. If each individual randomly responds the survey question with a biased coin flip of $p = 3/4$, the mechanism achieves $\ln 3$ local differential privacy.

The Warner model mentioned above is simple but influential. Some variations and extensions were developed, including the *Mangat Model* [40] and the *forced alternative response* [41]. Some other perturbation mechanisms such as *information compression* and *distortion* were also employed by different applications [42, 43].

3.2.3 Composition

As mentioned in Section 2.4, sequential composition and parallel composition are employed to provide a differentially private solution to a complex problem that involves more than one queries. By definition, centralized differential privacy is based on "neighboring datasets" and local differential privacy is defined on any two records of a dataset. The forms of privacy guarantee are the same. Therefore, local differential privacy inherits the sequential and parallel composition features mentioned in Section 2.4.

4 Privacy Attacks and Models of Differential Privacy for Social Networks

In this section, we first summarize the popular privacy attacks and provide insights on how to model privacy in social networks. Then we present four models of differential privacy, including node privacy, edge privacy, out-link privacy and partition privacy, which contribute to adapt differential privacy from tabular data to social network data,

4.1 Privacy Attacks in Social Networks

Privacy attacks [13, 44–48] refer to a wide variety of activities that leak sensitive information to unauthorized parties who should not know the information. The most serious type of privacy attacks in online social networks is inference attacks [49], which breach users' private information by analyzing background knowledge, such as user occupations or salary. Two classes of inference attacks are observed in social networks, namely private attribute inference [47, 48, 50–54] and user de-anonymization [45, 46, 55–61].

Private attribute inference aims to reveal a hidden attribute value that is intentionally protected by the user or service provider. Neighbor-based inference attacks [47, 48, 50, 51] abuse the fact that adjacent users may have the same or similar attribute values with a high probability and infer the private attribute of one user by exploiting the known attribute values of some other users sharing similar interests [53]. For example, if the majors of more than half of a user's friends are "computer science", then the user has a high probability of majoring in "computer science". Behavior-based inference [52–54] tries to identify the similarities of certain attribute values through the behavioral data, such as interests, characteristics and cultural behaviors. For example, if most of the apps, books, and music that a user likes are from China, there is a high probability that the user was originally from China.

User de-anonymization [45, 46, 55–61] takes an anonymized graph and a reference graph having the true user identities as inputs and maps the nodes in these two graphs such that the identities of the users in the anonymized graph can be reidentified. An anonymized social network graph is usually released by a service provider to various requesters, such as researchers, advertisers, app developers and government agencies, after hiding private identifiable information by various anonymization techniques, such as pseudonym, graph modification, clustering and generalization. A reference graph can be easily obtained through the gathered information from other sources such as a different social network which has overlapping users with a published social graph. Typically, a reference graph may have less attributes about nodes than an anonymized social network graph.

These two categories of privacy attacks lead to the exposure of different sensitive information. To protect private data in a social network and formalize the notion of "privacy" in social networks, distinctive privacy threats were recognized, which are shown in Figure 5.

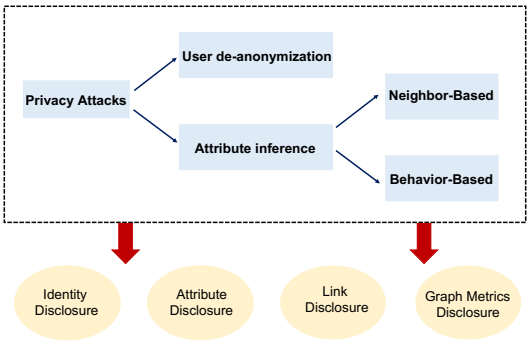


Fig. 5. Privacy attacks in online social networks.

- **Identity disclosure [13, 62]:** In social networks, the identity of an individual may be considered private, while attackers may exploit various user information to reidentify a social network user or to determine whether or not a target individual is present in a social network. For instance, AOL released an anonymized partial three-month search history to the public in 2006. Although personally identifiable information was carefully processed, some identities were accurately reidentified – *The New York Times* immediately located the following individual: the person with number 4417749 was a 62-year-old widowed woman who suffered from some diseases and had three dogs.
- **Attribute disclosure [13, 44, 63–65]:** A social network user's profile usually includes various attributes such as age, gender, major and occupation, some of which, such as salary, health status and disease information, are considered sensitive and private.
- **Link disclosure [13, 44, 62]:** The social relationships between individuals can be modeled as edges in a social graph. The link information may be considered sensitive in some cases. For example, Kossinets and Watts [66] analyzed a graph derived from email communications among students and faculty members in a university, of which the email relationships of "who emailed whom" was deemed sensitive [19].
- **Graph Metrics disclosure [16]:** Since social networks can be modeled as graphs, graph metrics, such as degree, betweenness, closeness centrality, shortest path length, subgraph counting and edge weight, may be employed to conduct social network analysis. The disclosure of such information may indirectly lead to privacy leakage. For example, many de-anonymization attacks are based on the structure information of a social graph.

Modeling privacy is critical for realizing privacy preservation in social networks. Differential privacy assumes the maximum background knowledge for adversaries. In the subsequent section, we present how to protect social networks under differential privacy by extending the differential privacy definition from traditional databases to graphs, and demonstrate how to formally define differential privacy in social networks based on the privacy threats mentioned above.

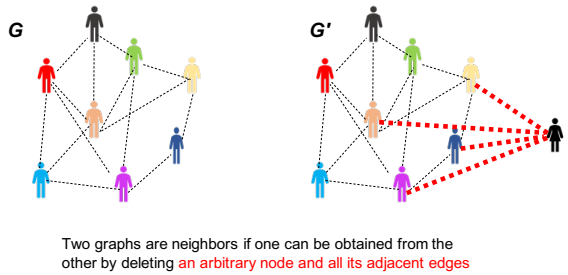


Fig. 6. Node privacy.

4.2 Differential Privacy Models for Social Networks

A social network can be modeled as a graph $G(V, E)$, where V is a set of nodes and E is a set of relational activities between nodes. Differential privacy originates from traditional databases. The key to extending differential privacy to social networks is to determine the neighboring input entries, that is, how to define "adjacent graphs". In this subsection, we review the applications of differential privacy in social networks by presenting adjacent graphs defined on node, edge, out-link and graph partition, and describing the corresponding privacy models of node privacy [34], edge privacy [34], out-link privacy [67] and partition privacy [68].

4.2.1 Node Privacy

A privatized query Q preserves *node privacy* [19] if it satisfies differential privacy for every pair of graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ such that $|(V_1 \cup V_2) \setminus (V_1 \cap V_2)| = 1$ and $\{(E_1 \cup E_2) \setminus (E_1 \cap E_2)\} = \{(u, v)|u = x \vee v = x\}$, where x is the only node in $(V_1 \cup V_2) \setminus (V_1 \cap V_2)$ and (u, v) represents the edge between nodes u and v.

In node privacy, an adjacent graph G' of a given social network G is the one obtained by deleting or adding a node and all edges incident to that node, as shown in Figure 6. Node differential privacy intends to prevent an attacker from determining whether or not an individual node x appears in the graph. It guarantees privacy preservation for individuals and relationships simultaneously rather than just a single relationship, at the cost of strict restrictions on queries and reduced-accuracy results. A differentially private algorithm must conceal the worst-case discrepancy between adjacent graphs, which may be substantial under node privacy. For example, if we consider an extreme case where a node connects to all other nodes (a star graph), then the sensitivity is high and the added noise has to be dramatic, too. Generally speaking, node privacy is infeasible to provide high utility (accurate network analysis) due to high sensitivity, but it provides desirable privacy protection [19].

4.2.2 Edge Privacy

A privatized query Q preserves *edge privacy* [19] if it satisfies differential privacy for each pair of graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ such that $V_1 = V_2$ and $|(E_2 \cup E_1) \setminus (E_2 \cap E_1)| = 1$.

In edge privacy, an adjacent graph G' of a given social network G is obtained by deleting or adding one edge from G, as shown in Figure 7. It can be generalized to allow at most k edges to be changed.

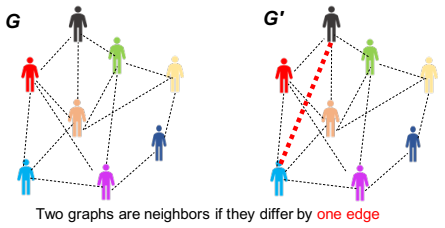


Fig. 7. Edge privacy.

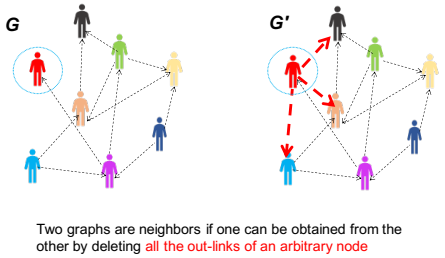


Fig. 8. Out-link privacy.

Edge privacy protects against learning about specific relationships between users and prevents an attacker from determining with a high certainty whether two individuals are connected. Comparing with node privacy, edge privacy can only provide protection on information about relationships between users. Nodes with higher degrees still have a higher impact on query results, despite the fact that the relationships between these nodes have been protected. Edge privacy provides meaningful privacy protection in many practical applications and has been more widely used than node privacy [67]. For example, Kossinets and Watts [66] employed edge privacy to protect email relationships.

4.2.3 Out-Link Privacy

A privatized query Q preserves *out-link privacy* [67] when it meets the definition of differential privacy for every pair of graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ such that $V_1 = V_2$ and there exists a node x such that $\{(E_1 \cup E_2) \setminus (E_1 \cap E_2)\} = \{(x \rightarrow v) | x \in V_1 \wedge v \in V_2 \text{ or } x \in V_2 \wedge v \in V_1\}$, where $(x \rightarrow v)$ is a directed link from x to v.

In out-link privacy, for a given social network G, an adjacent graph G' is obtained by either removing all the existing out-links of a node x, or adding one or more new out-links to a node whose out-degree in G is 0. See Figure 8 for an example. It protects the out-links of a node using the same conceptual privacy standard as node privacy.

Out-link privacy can reduce the distinguishing properties of high-degree nodes, that is, a high-degree node can deny that the friendships are mutual in query results although others claim to be friends with this node. Out-link privacy is strictly weaker than node privacy, but for certain query functions it has better performance than edge privacy [68]. Out-link privacy simplifies the calculation of sensitivity and reduces the amount of injected noise required, thus allowing certain queries that are infeasible under node privacy and edge privacy [67]. In Section 5.1 we take the degree distribution as an example to demonstrate that the out-link privacy requires less noise.

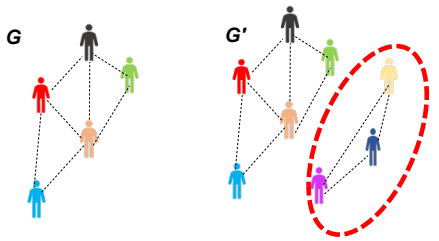


Fig. 9. Partition privacy.

4.2.4 Partition Privacy

A partitioned graph G is comprised of multiple disjoint components H_i [68]. A privatized query Q preserves *partition privacy* if it satisfies differential privacy for every pair of graphs G_1 and G_2, where $G_1 = G_2 - H_i$ with $H_i \in G_2 \wedge H_i \notin G_1$ or $G_2 = G_1 - H_j$ with $H_j \in G_1 \wedge H_j \notin G_2$.

In partition privacy, an adjacent graph of a given social network G is obtained by adding a new or deleting an existing subgraph from G (see Figure 9 for an illustration). Most social-structure queries are conducted over a set of subgraphs instead of a connected social graph. Some attributes of the nodes such as address, major, and education level can be used to partition a large social graph into multiple subgraphs, and each subgraph can be treated as a multi-attribute data point. Then, deleting or inserting a subgraph is equivalent to removing or adding a data point [68]. Accordingly, traditional differential privacy can be applied to the set of subgraphs (data points).

Partition privacy provides broader preservation than node privacy, and the protection is applied not to a single node, but to a social group.

5 Differential Privacy in Social Network Analysis

In this section, we summarize the state-of-the-art research on a series of most-widely used differentially private social network analysis techniques. Social network analysis refers to the quantitative analysis on the data generated by social network services using statistics, graph theory and other techniques. Some popular tasks of social network analysis include degree distribution, subgraph counting (triangle counting, k-star counting, k-triangle counting,etc.) and edge weight analysis. In this section, we analyze a few widely used techniques in social network analysis under differential privacy preservation. Table 4 summarizes the major existing differentially private social network analysis techniques for degree distribution and subgraph counting while those for edge weight is summed up in Table 5.

5.1 Degree Distribution

Degree distribution is one of the most widely studied graph characteristics. It reflects the graph structure statistics and may affect the whole process of graph operations. Degree distributions can be employed to describe the basic social network structures, design graph models and measure graph similarities.

TABLE 4
Summary on Existing Differentially Private Graph Analysis Techniques

Graph statistics	Privacy standard			
	Edge privacy	Node privacy	Out-link Privacy	Partition Privacy
Degree distribution	[19], [69]	[21], [20], [70–73]	[68]	[68]
Graph publishing/sharing	[74], [23]			
Triangle counting, Centrality			[68]	[68]
Triangle counting, Clustering, MST cost	[22]			
Cut function of graph	[75]			
Subgraph counting	[34], [76], [24], [77] (Weaker than edge privacy)	[21], [71], [24]		
Average degree, Distance to connectivity		[78] (Stronger than node privacy)		

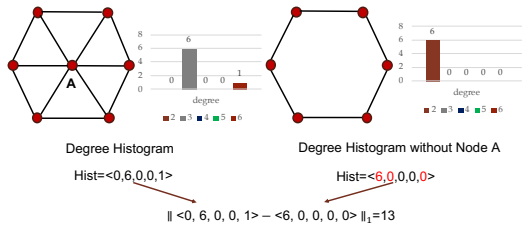


Fig. 10. Degree histogram under node privacy.

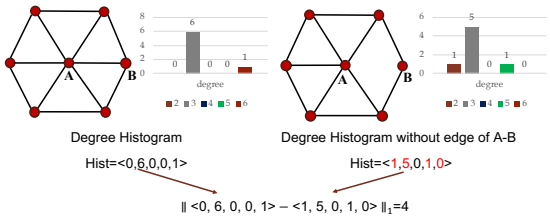


Fig. 11. Degree histogram under edge privacy.

The degree distribution of a graph can be simply transformed to a degree sequence by counting the frequency of each degree. Here we use a degree histogram to describe the degrees of the nodes in a graph. Consider the example shown in Figure 10. One can see that the degree counts change significantly when deleting node A. This implies that the sensitivity of degree distribution is high under node privacy since the change of one node may affect multiple degree counts. A careful analysis reveals that a node of degree k affects $2k + 1$ values of the histogram at most. In the worst case, the addition or deletion of a node of the maximum degree results in the change of $2n + 1$ values, which indicates that the global sensitivity depends on the value of n, the number of nodes in the graph. Since n is unbounded, the degree histogram (distribution) query is not feasible for differential privacy protection under node privacy.

Under edge privacy, protecting degree histogram queries using differential privacy is feasible, as illustrated in Figure 11. One can see that removing an edge from a network only changes the degrees of two nodes, thus affecting 4 counts at most. The sensitivity is $4k$ under the k-edge privacy. Accordingly, when k is small, the amount of added noise is relatively small and even negligible for a graph that is large enough, providing preservation in data utility.

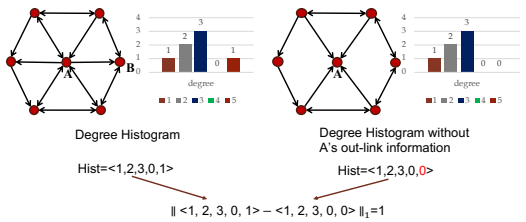


Fig. 12. Degree histogram under out-link privacy.

Out-link privacy requires less noise for a degree histogram query. Removing out-links of one node from a graph affects one value in the histogram when only out-degrees are counted, as shown in Figure 12. Under the out-link privacy, although one node may delete all its out-links from the graph, this node can still be identified by its friends' out-degrees. Nevertheless, a slightly higher-than-expected node degree in a graph may not be easily identified [68]. Therefore, if an attacker intends to guess the presence of a high-degree node with certainty, she may have to learn full knowledge about the social network.

To obtain differentially private results in degree distribution analysis, a number of techniques were proposed, such as *post-processing* [19, 69], *projection* (also known as *bounded degree*) [20, 21, 70, 71], *Lipschitz extension* [72], *Erdös-Rényi graph* [73] and *random matrix projection* [74]. Post-processing and projection are the most commonly used ones.

5.1.1 Post-Processing Techniques

Hay *et al.* [69] proposed a post-processing technique to boost the accuracy of the existing differentially private algorithms. The key idea is to find a new set of answers that is the "closest" to the set of noisy ones returned from differentially private algorithms by means of "constrained inference" for better accuracy, that is, enforcing consistency constraints among the noisy query results. It involves three steps. First, an analyst sends to the data owner a set of queries with constraints holding among the corresponding answers for a given task. Then the data owner replies to the set of queries using standard differentially private algorithms. In the third step, the analyst post-processes the set of noisy answers with constrained inference to resolve the possible inconsistencies among the noisy answers for the purpose of finding a new set of answers that is the closest to the old one while satisfying the consistency constraints. Here, "closest" is measured in L_2 distance, and the result is a minimum

L_2 solution. This technique can be viewed as an instance of linear regression.

We use an example to illustrate the procedure. Suppose an analyst needs answers to the total number of students x_t, the numbers of students x_A, x_B, x_C, x_D and x_F, respectively, receiving grades A, B, C, D and F, and the number of passing students x_p, from a private student database. Intuitively the analyst can obtain differentially private answers to $(x_A, x_B, x_C, x_D, x_F)$, and then use them to compute those for x_t and x_p. Nevertheless, based on the post-processing approach proposed in [69], the analyst first requests differentially private answers to all queries $x_t, x_p, x_A, x_B, x_C, x_D, x_F$, then applies the two constraints $x_t = x_p + x_F$ and $x_p = x_A + x_B + x_C + x_D$ to derive more accurate answers for x_t and x_p. Hay *et al.* [69] claimed that the above post-processing technique does not sacrifice privacy.

Hay *et al.* [19] adapted the definition of differential privacy to graph-structure data and proposed a differentially private algorithm based on the post-processing technique proposed in [69] to obtain an approximation of a graph's degree distribution. The authors provided the minimum L_2 solution to the degree distribution query. The basic idea is to obtain the query results of a graph's degree sequence in a non-decreasing order, then transform them to a degree distribution by counting the frequency of each degree. Let S denote the degree sequence query $S = \langle deg(1), \ldots, deg(n) \rangle$, of which $deg(i)$ denotes the i^{th} smallest degree in G. For example, assume that the degrees of a five-node graph are $\{3, 3, 3, 2, 1\}$, then $S = \langle 1, 2, 3, 3, 3 \rangle$. Let $\tilde{S}$ denote the sorted results of the differentially private algorithm seeking the degree of each node. Since the degrees are positioned in a sorted order, S is constrained, which can be denoted by $S[i] \leq S[i+1]$ for $1 \leq i < n$. Then the minimum L_2 solution $\bar{S}$ is obtained by applying constrained inference to $\tilde{S}$. Considering the example, if the differentially private result is $\tilde{S} = \langle 1, 9, 4, 3, 4 \rangle$, the algorithm computes the minimum L_2 solution $\bar{S}$ as $\langle 1, 5, 5, 5, 5 \rangle$ based on the constraints of $S[i] \leq S[i+1]$.

5.1.2 Bounded Degree Techniques

Kasiviswanathan *et al.* [21] proposed a carefully-designed projection scheme mapping an input graph to a bounded degree graph to obtain the degree distribution of the original one under node privacy. Aiming at obtaining statistical information with low sensitivity, the original network is projected to a set of graphs whose maximum degree is lower than a certain threshold. In a bounded degree graph, node privacy is easier to achieve as the sensitivity can be much smaller for a given query function. When the degree threshold is carefully chosen for realistic networks, such a transformation leaks little information. Two families of random distributions are adopted for the noise: Laplace distributions with global sensitivity and Cauchy distributions with smooth sensitivity. The key difficulty of this approach lies in that the projection itself may be sensitive to the change caused by a single node in the original graph. Thus, the process of projection should be "smooth" enough to ensure the privacy-preservation property of the entire algorithm. Two different techniques were proposed in [21]. The first one defines *tailored projection operators*, which have low sensitivity and protect information for specific statistics. The second one is a "naïve" projection that just simply discards the high-degree nodes in a graph. Interestingly, the naïve projection enables the design of algorithms to bound the local sensitivity of the projected graph, and the development of a generic reduction technique allows differentially private algorithms to be applied to bounded-degree graphs.

Day *et al.* [20] proposed an edge-addition based graph projection method to reduce the sensitivity of the graph degree distribution under node privacy. This improved projection technique preserves more information than previous ones. It was proved in [20] that the degree histogram under the projected graph has sensitivity $2\theta + 1$ for a θ-bounded graph in which the maximum degree is θ. Based on this sensitivity bound, two approaches, namely (θ, Ω)-Histogram and θ-Cumulative Histogram, for degree histograms were proposed under node privacy. Macwan *et al.* [70] adopted the same method of edge-addition [20] to reduce the sensitivity of the node degree histogram. Note that existing projection-based approaches cannot yield good utility for continual privacy-preserving releases of graph statistics. To tackle this challenge, Song *et al.* [71] proposed a differentially private solution to continually release degree distributions with a consideration on privacy-accuracy tradeoff, assuming that there is an upper bound on the maximum degree of the nodes in the whole graph sequence.

5.1.3 Other Techniques

Raskhodnikova *et al.* [72] proposed an approximation of the graph degree distribution by making use of the Lipschitz extension and the generalized exponential mechanism under node privacy. Sealfon *et al.* [73] developed a simple, computationally efficient algorithm for estimating the parameter of an Erdös-Rényi graph under node privacy. This algorithm optimally estimates the edge-density of any graph whose degree distribution is concentrated on a small interval. Ahmed *et al.* [74] presented a random matrix approach to social network data publishing, which achieves differential privacy with storage and computational efficiency by reducing the dimensionality of adjacency matrices with random projection. The key idea is to first randomly project each row of an adjacency matrix into a low-dimensional space, then perturb the projected matrix with random noise, and finally publish the projected and perturbed matrix. The random projection retains the graph matrix's top eigenvectors. As both random projection and random perturbation can preserve differential privacy with a small amount of noise, data utility can be improved.

5.2 Subgraph Counting

Given an input graph G and a query graph H, a subgraph counting query asks for the number of isomorphic copies of H in G. Example subgraphs include triangles, k-triangles, k-stars, and k-cliques, where a k-triangle consists of k triangles sharing one common edge, a k-star is composed of a central node connecting to k other nodes, and a k-clique is a clique with k vertices. Figure 13 demonstrates these subgraphs.

Note that subgraph counting counts the copies of a subgraph. Therefore a node of degree $d \geq k$ contributes $\binom{d}{k}$ to k-star counting. Figure 14 presents a few examples of

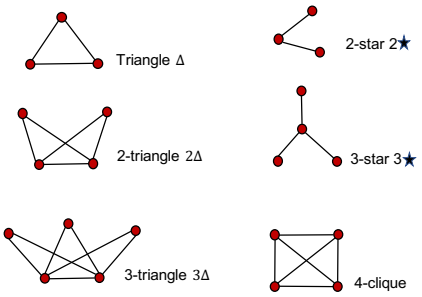
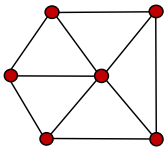


Fig. 13. Examples of subgraphs.



Subgraph	Count
Triangle	5
2-star	25
3-star	15
2-triangle	5
3-triangle	0

Fig. 14. Examples of subgraph counting.

subgraph counting. We consider the counting problems of triangle, k-star, and k-triangle in this section, and denote them respectively by $f_\triangle$, f_{k*} and $f_{k\triangle}$. These counting results are keys to many descriptive graph statistics that are used to describe and compare graph properties and structures. For example, the clustering coefficient of a graph is the ratio of $3f_\triangle$ over f_{2*}.

Subgraph counting queries generally have different privacy characteristics and high global sensitivities. To realize differential privacy, it is necessary to add a large amount of noise, which may lead to serious query result distortions. Therefore, a smooth upper bound of the local sensitivity is usually used to determine the noise magnitude. Additionally, truncation, Lipschitz extension and ladder function were adopted in literature [21, 34, 76] to achieve differential privacy while improving the counting performance.

Before summarizing the state-of-the-art techniques, let us introduce the following notations. For an undirected graph with n nodes, the adjacency matrix is $X = (x_{ij})$, where $x_{ii} = 0$ for all $i \in [n]$. Let a_{ij} denote the number of common neighbors shared by a particular pair of vertices i and j, that is, $a_{ij} = \sum_{l\in[n]} x_{il} \cdot x_{lj}$. Let b_{ij} denote the number of vertices connected only to one of the two vertices i and j, that is, $b_{ij} = \sum_{l\in[n]} x_{il} \bigoplus x_{lj}$. Denote by $d(G, G')$ the distance between two n-vertex graphs G and G', which is the number of edges they differ. Graph G and G' are neighbors if $d(G, G') = 1$. Let $LS_\triangle$, $LS_{k\triangle}$ and LS_{k*} denote the local sensitivities of $f_\triangle$, $f_{k\triangle}$ and f_{k*}, respectively. Denote by $S^*_{\triangle,\beta}$, $S^*_{k\triangle,\beta}$ and $S^*_{k*,\beta}$ the smooth sensitivities of $f_\triangle$, $f_{k\triangle}$ and f_{k*}, respectively.

5.2.1 Triangle Counting

As mentioned earlier, node privacy is a strong privacy guarantee, so it is not feasible to obtain a triangle counting satisfying node privacy in most cases. At the worst case, adding a vertex to a complete n-node graph brings $\binom{n}{2}$ new triangles. Since this change depends on the size of the graph, the global sensitivity of triangle counting is unbounded. Moreover, triangle counting is also not feasible under edge privacy as in the worst case, deleting one edge from an n-node graph deletes $n-2$ triangles. Although the global sensitivity of a triangle counting query is not bounded, its local sensitivity for some specific graphs is bounded under edge privacy. Thus, smooth sensitivity [22, 34] can be adopted to achieve differential privacy. In the following, we briefly summarize edge and node differentially private algorithms as well as other techniques to achieve differential privacy in triangle counting.

5.2.1.1 Edge Differentially Private Algorithms: Nissim *et al.* [22] introduced an approach to calculate the smooth sensitivity of triangle counting and provided the cost of a minimum spanning tree under edge privacy. The local sensitivity of $f_\triangle$ is $LS_{f_\triangle} = \max_{i,j\in[n]} a_{ij}$, the global sensitivity is $GS_{f_\triangle} = n-2$, while $LS_{f_\triangle}$ at distance s is $LS^{(s)}_{f_\triangle} = \max\limits_{i\neq j;i,j\in n} c_{ij}(s)$, where $c_{ij}(s) = \min(a_{ij} + \lfloor\frac{s+\min(s,b_{ij})}{2}\rfloor, n-2)$. The β-smooth sensitivity of $f_\triangle$ has time complexity $O(M(n))$, where $M(n)$ is the time required for multiplying two matrices of size $n \times n$.

Karwa *et al.* [34] presented an efficient algorithm for outputting approximate answers to subgraph counting queries, such as triangle counting, k-star counting and k-triangle counting. These algorithms satisfy edge privacy and can be regarded as an extension of the algorithm in [22] to a bigger class of subgraph counting problems with privacy guarantees and better accuracy.

Sala *et al.* [23] proposed a differentially private graph model called *Pygmalion* to generate synthetic graphs. They adopted the dK-graph model and its statistical series as the query function. The dK-graph model extracts the detailed structure of a graph into degree correlation statistics, and outputs a synthetic graph using the dK-series values. A dK-series is the degree distribution of connected components of certain size within a target graph. Here, the dK-series is a graph transformation function. Sala *et al.* [23] first proved that the dK-series has a high sensitivity, then proposed a partitioning approach to group tuples with similar degrees, which effectively reduces the noise magnitude and achieves a desired privacy guarantee.

Zhang *et al.* [76] proposed an approach of specifying a probability distribution over possible outputs to maximize the utility of an input graph while providing a privacy guarantee. They applied a ladder function to the subgraph counting problems of triangle, k-star and k-clique, and achieved high accuracy with efficient time complexities.

Gupta *et al.* [75] considered the problem of approximately publishing the cut function of a graph under edge privacy. They proposed a generic framework of converting iterative database construction algorithms into privatized query publishing approaches under non-interactive and interactive settings.

5.2.1.2 Node Differentially Private Algorithms: Since node privacy is a strong privacy guarantee, a large amount of noise needs to be added, leading to a dramatic distortion of the graph structure and a poor utility. One of the most widely adopted mechanisms is the generic reduction to privacy over a bounded-degree graph. If a graph is known to have a maximum degree of d, deleting or adding a node may affect $\binom{d}{2}$ triangles at most. For

graphs whose maximum degree is greater than d, high-degree nodes can be deleted to get a graph with a maximum degree falling within a threshold. The number of triangles of this bounded-degree graph can be a good approximation to the true query answer. Therefore, networks with a small number of large degree nodes can adopt this approach to achieve node privacy for triangle counting.

Kasiviswanathan *et al.* [21] proposed algorithms for releasing statistics of graph data under node privacy. On the basis of smooth sensitivity of truncation, they presented a generic reduction mechanism in order to apply differentially private algorithms for bounded-degree graphs to arbitrary graphs, that is, just simply removing the nodes with high degrees. A continual privacy-preserving release of subgraph counting under node privacy was investigated in [71], which assumes that there is a publicly known upper bound on the maximum degree of the nodes in the graphs.

Blocki *et al.* [24] proposed the definition of *restricted sensitivity*, which can improve the accuracy of differential privacy compared with global sensitivity and smooth sensitivity. Two important query classes, namely subgraph counting and local profile matching of social networks, were analyzed. It was proved that the restricted sensitivities of these two kinds of queries are much lower than those under smooth sensitivity. More importantly, when computing the smooth sensitivity involves higher computational complexity and lower efficiency, restricted sensitivity performs better.

5.2.1.3 Other Types of Privacy: Rastogi *et al.* [77] considered general privacy-preserving social network queries including subgraph counting. They proposed a relaxation of edge privacy, called a *theoretic standard of adversarial privacy*. Their algorithm can release more general graph statistics than the algorithms in [22], which only deal with triangles. However, the assumption on adversarial privacy puts some limits on the applicability of this privacy definition [77].

Task *et al.* [68] proposed two differential privacy standards, i.e., *out-link privacy* and *partition privacy*, over network data. They also introduced two algorithms respectively satisfying the two privacy standards to release approximate results of degree distribution query, triangle counting and centrality counting. It was demonstrated that partition privacy can provide stronger privacy guarantee with less noise when cross-analyzing multiple social networks.

Gehrke *et al.* [78] presented a zero-knowledge based privacy definition, which is stronger than differential privacy. They constructed a zero-knowledge private mechanism to release the social graph structure information such as the average degree and the distance to connectivity.

5.2.2 k-Star Counting

Karwa *et al.* [34] extended the approach in [22] to the k-star counting query and proposed how to compute the local sensitivity and smooth sensitivity of f_{k*}. They proved that these two sensitivity values of k-star counting are equal, that is, $S^*_{k*,\beta}(G) = LS_{k*}(G)$ when $d_{\max} \geq \max\{k, (k-1)(\frac{1-\beta}{\beta})\}$, of which $d_{\max}$ is the largest degree in G.

Kasiviswanathan *et al.* [21] proposed an (ϵ, δ)-node differentially private algorithm with a linear programming (LP) based function for the special case of the subgraph H having 3 nodes, e.g., H can be a triangle or a 2-star. If $f_H(G)$ (the number of copies of H in G) is relatively large, the Laplace mechanism provides an accurate estimate. The release of $f_H(G)$ is more accurate with the LP-based function when $f_H(G)$ is smaller.

Zhang *et al.* [76] presented a ladder function and applied it to the k-star query under edge privacy. The ladder function relies on a carefully designed probability distribution that can maximize the probability of outputting true answers and minimize that of outputting the answers that are far from the true answers. In addition, to achieve differential privacy, it is constrained that the probabilities of outputting a value for the input graph g and its neighbor g' should be very close. The authors adopted the concept of "local sensitivity at distance t" in [21] to create a ladder function. In fact, the upper bound of the "local sensitivity at distance t" was used as the ladder function for the f_{k*} query.

5.2.3 k-Triangle Counting

When triangle counting is extended to k-triangle counting, the problem becomes complicated as it is NP-hard to calculate the smooth sensitivity of k-triangle counting. Therefore, existing approaches mainly focus on a small k, while the counting query of $f_{k\triangle}$ itself is hard.

An approach was proposed in [34], whose main idea is to compute (ϵ, δ)-differential privacy (edge privacy) by adding noise proportional to a second-order local sensitivity instead of a "smooth" upper bound. Since $LS_{k\triangle}$ cannot be directly adopted with the Laplace mechanism, LS', the local sensitivity of $LS_{k\triangle}$, was employed. It was demonstrated that LS' is a deterministic function of a quantity with global sensitivity 1, based on which the query results can be published with less noise. Another approach was presented by Zhang *et al.* [76], which provided a ladder function for k-triangle counting under edge privacy.

5.3 Edge Weights

In social networks, social relations are modeled on edges with weights. An edge may reveal different sensitive information between individuals, such as the communication cost, the interaction frequency between two social network users, the price of a commercial trade or the similarity between two organizations. Thus, releasing edge weights must be done in a privacy preserving manner. Table 5 summarizes the most popular exiting differentially private edge weight algorithms in social networks.

Liu *et al.* [79] studied the problems of protecting privacy in edge weights and preserving the utility of statistics of shortest paths between nodes. They proposed two edge privacy-preserving approaches, namely *greedy perturbation* and *Gaussian randomization multiplication*. The former mainly focuses on preserving the length of the perturbed shortest paths and the latter retains the same shortest paths before and after perturbation.

Das *et al.* [80] conducted edge weight anonymization in social graphs. They developed a linear programming model to protect graph characteristics such as shortest paths, minimum spanning trees and k-nearest neighbors, which can be formalized as linear functions of the edge weights.

TABLE 5
Summary on Existing Edge Weight Preservation Techniques

	Approach
[79]	Gaussian randomization multiplication; Greedy perturbation
[80]	Linear programming model
[81]	Edge weight-count; Laplace perturbation
[82]	Edge weight-unattributed histogram; k-indistinguishability

Costea *et al.* [81] considered differential privacy protection to the edge weights assuming that the graph structure is public and available to users without modification while the edge weights are private. They employed the Dijkstra algorithm to get the shortest paths for protection quality evaluation.

Last, Li *et al.* [82] treated the edge-weight sequence as an unattributed histogram by merging all barrels with the same count into one group and thus ensured k-indistinguishability among groups. They proposed an approach with Laplace noise added to every edge weight to improve accuracy and utility of the published data.

5.4 Summary

Social networks contain information about social users, their attributes as well as social relationships, which are usually deemed sensitive. The release of such information may bring significant privacy concerns or even damages to personal reputation and properties if the protection on sensitive information is not sufficiently strong. In this section, we discussed degree distribution, subgraph counting (triangle, k-star and k-triangle) and edge weights, the most popular graph analysis techniques in social networks. Note that there exist other statistics on graph structures but the basic methods and ideas of adopting differential privacy are similar and thus are omitted here.

Most existing differentially private algorithms have to pay a substantial tradeoff in utility (e.g., accuracy) for privacy preservation in analyzing large-scale and complex graph structures. Indeed, many of those methods try hard to improve utility as their major contributions. Moreover, the complexities of the differentially private algorithms are generally high or even NP-hard due to the complexity of computing (smooth) sensitivities. In some cases such as k-triangle counting, even the structure query itself is NP-hard.

6 Local Differential Privacy in Social Network Analysis

Most of the studies reviewed in Section 5 adopt centralized differential privacy in order to provide strong privacy guarantee. As analyzed above, the privacy preservation comes with a tradeoff in data utility. Local differential privacy (Definition 8 [38]) presents an important alternative to balance the tradeoff between privacy preservation and utility by exploring local data-dependent mechanisms. In this section, we review the new progresses on applications of local differential privacy in social network analysis. Table 6 summarizes a series of research problems including statistical databases and social network analysis that are related to local differential privacy.

Local differential privacy has mainly been applied to statistical databases for problems like frequency statistics publishing, which involves frequency publishing for discrete data [84] and mean value publishing for continuous data [38, 88, 91]. Frequency distributions can be expressed by contingency tables, histograms, and some other forms. According to the number of variables, frequency distributions can be divided into single-valued ones [83–87] and multi-valued ones [88–90]. To protect privacy in single-valued frequency distributions, local differential privacy can be achieved by performing perturbation, such as the random response mechanism [39], directly on the encoded values or the hashed values, and then conducting aggregation and estimation. For the latter case, sampling and dimension reduction techniques are often used to improve data utility. In [92], Cheu *et al.* developed a shuffled model for distributed differentially private algorithms, which can be regarded as a technique that is between centralized differential privacy and local differential privacy.

Local differential privacy can also be applied to social networks as each social user has a local graph and an overall social graph can be generated based on all users' local graphs. A number of interesting applications of local differential privacy were reported recently. For example, Qin *et al.* [25] made an effort to ensure individual's local differential privacy while gathering structural information to generate synthetic social graphs. They proposed a multi-phase technique named LDPGen, which incrementally clusters structurally similar users via refining parameters into different partitions. Specifically, whenever a user reports information, LDPGen deliberately injects noise to guarantee local differential privacy. Moreover, LDPGen derives optimal parameters to cluster structurally similar users together. After obtaining a good clustering, LDPGen constructs a synthetic social network by adopting the existing Chung-Lu social graph generation model [95].

Sun *et al.* [93] pointed out that it is insufficient to apply local differential privacy to protect all network participants when collecting extended local views (ELV). The main problem lies in that each individual has its own local privacy budget, which covers its own ELV regardless of its neighbors and the specific information from its ELV. To prevent this attack, a novel decentralized differential privacy (DDP) mechanism was proposed, which demands each participant to consider not only its own privacy, but also those of the neighbors in its ELV. Towards this goal, a multi-phase mechanism under DDP was developed, which allows an analyst to better estimate subgraph counting. In this framework, an analyst first queries each individual's minimum noise scale, which must be performed under DDP since it relies on the local graph structure and is private. Then, the analyst calculates the minimum noise scale for the whole network and gathers subgraph counts accordingly.

Ye *et al.* [96] presented an LDP-enabled graph metric estimation framework LF-GDPR for graph analysis. LF-GDPR first collects the adjacency bit vector and node degree from each node locally; then provides the perturbation protocols and the aggregation and calibration algorithms

TABLE 6
Summary on Existing LDP-Based Techniques

Research Problems	Results
Perturbation mechanism	Compression [42], Distortion [43]
Single-valued frequency distribution	S-Hist [83], O-RR [84], PCE [85], k-Subset [86, 87]
Multi-valued frequency distribution	Harmony-frequency [88], PARROR-Unknown [89], LoPub [90]
Mean value publishing	Minimax rate [38], MeanEst [91], Harmony-Mean [88]
Distributed differential privacy	Shuffled model [92]
Synthetic social graph generation	LDPGen [25] (Edge-LDP)
Subgraph counting (triangles, three-hop paths, k-clique)	ELV based on decentralized DP [93] (Stronger than Edge-LDP)
Attributed graph data	AsgLDP [94] (Edge-LDP)

for the two graph analysis task. Wei *et al.* [94] presented a novel framework AsgLDP to collect and generate privacy-preserving attributed graph data that satisfies LDP. AsgLDP first collects the aggregate information of the original decentralized attributed graph, then optimizes the privacy-utility tradeoff of the generated data to preserve general graph properties such as attribute distributions, degree distributions and community structures.

To the best of our knowledge, no high sensitivity problem was reported in local differential privacy. However, it is a great challenge for data collectors to reconstruct a graph structure with high utility based on the disturbed or local graph data, i.e., to ensure the preservation of correlations between different users' local graph data when the perturbation process of each user is independent of each other. Furthermore, if we only collect graph statistics, such as node degrees and subgraph counting, to generate composite graphs, an output graph may not retain the important characteristics such as connectivity, clustering coefficient, closeness centrality or even graph structure of the original one, and thus may reduce graph utility.

7 Conclusions and Future Directions

In this article, we provide a survey on differential privacy foundations and applications in protecting the privacy of social network analytical results. We explain the underlying design principles of different mechanisms and present the state-of-the-art research results. To achieve differential privacy, one needs to specify a privacy budget and calculate the amount of noise to be added to the query results. The privacy budget determines the level of privacy preservation: the smaller, the better the protection. At the same time, the noise magnitude affects the accuracy (utility) of the query results, which should be minimized provided that sufficient privacy protection is achieved. Noise magnitude is derived from sensitivity and privacy budget. When global sensitivity is high, smooth sensitivity may be employed instead.

The research on differential privacy is developing fast, and its applications in social network analysis enjoy stronger and stronger interest from industry and academia. In the following we discuss a few open research problems in differential privacy technologies for social network analysis.

7.1 Differential Privacy for Complex and Correlated Social Network Data

In social networks a user often has relationships with many others at different levels. Thus, network structures are often complex. Since query sensitivities in social networks are usually high, much noise has to be added to query results to achieve differential privacy. Nevertheless, the noise may significantly affect the output data utility. In addition, it may be hard to effectively compute sensitivities, either global or smooth, precise or approximate, as the computational complexity may be too high (or even NP-hard) to be practical for many complex social network analysis queries. Even though a large number of studies reviewed earlier focus on how to apply differential privacy to complex social structure queries, most of them are limited to "small" queries, such as a small k in k-star and k-triangle counting. It remains a great challenge to employ traditional differential privacy for complex graph queries.

Moreover, in social networks, social correlations are usually strong as behaviors and attributes of adjacent nodes are often strongly related. For example, adjacent users may have the same attributes with a high probability. Therefore, the private attributes of a social network node may be inferred by exploring the publicized attributes of its neighbors which share common interests [53]. The social relations, that is, the edges in a social network, are often not independent, as the social relationship between two nodes may depend on a third node that is a common neighbor. To address dependency in data, dependent differential privacy has attracted a lot of attention in recent years [26, 37]. Nevertheless, applying dependent differential privacy to social networks remains to be a grand open challenge due to high dependencies and complex social structures.

To tackle the challenges, one possible direction is the transformation techniques. For example, we may consider adding a sampling process to transform an original graph data to one in a different domain such that the data tuples become independent and sparse and thus traditional differential privacy can be applied. This is motivated by the random but uniform sampling step in [97]. The non-uniform compressive sampling technique [98–100] may be employed as it can realize the required transformation with controlled distortions.

7.2 Tradeoff between Privacy Budget and Data Utility

How to allocate an appropriate amount of privacy budget to achieve sufficient privacy protection on sensitive data and, at the same time, maximize data utility remains a fundamental challenge [101]. Recently various schemes were developed to investigate the privacy-utility tradeoff based on techniques such as game theory and linear programming [102–105].

Dwork *et al.* [105] stated that there is little understanding on the optimal value of privacy budget for a practical scenario. Importantly, their interview results obtained from surveying different practitioners regarding how organizations made key choices when implementing differential privacy in practice indicated that there was no clear consensus on how to choose privacy budget, nor agreement on how to approach to the problem. One challenge is to quantify the tradeoff between privacy budget and data utility.

7.3 Differentially Private Publishing of High Dimensional Social Network Data

The unprecedented growth and popularity of online social networks have generated massive high-dimensional data, such as social users' attribute information, healthcare data, location information, trajectory data, and commercial electronic data, which is often published or made available to third parties. However, publishing such attribute data may disclose private and sensitive information and result in increasing concerns on privacy violations. Differentially private publishing of such data has received broad attentions. Nevertheless, most differentially private data publishing techniques cannot work effectively for high dimensional data. On one hand, since the sensitivities of different dimensions vary, evenly distributing the total privacy budget to each dimension degrades the performance. Moreover, the "Curse of Dimensionality" leads to two critical problems. First, a dataset containing many dimensions and large attribute domains has a low "Signal-to-Noise Ratio" [106]. Second, complex correlations exist between attribute dimensions, making it impossible to directly and independently protect each dimension's privacy. To address these challenges, one may conduct data dimensionality reduction. However, it is hard to maintain the characteristics of high dimensional data to the maximum extent and to prevent private information from being defected during the process of dimensionality reduction.

To address these challenges, Bayesian networks [106], random projection [107] and various sampling techniques were employed to support differentially private high dimensional data publishing [108, 109]. Nevertheless, most of these approaches still cannot work effectively for releasing high-dimensional data in practice as they generally ignore the different roles a dimension may play for a specific query – one dimension may be more important than another for a particular query. Additionally, one dimension may release more information than another if the same amount of noise is added. Therefore, how to allocate the total privacy budget to dimensions is query-dependent and should be carefully investigated. Moreover, the underlying distribution of the data may be unknown and the high dimensionality and large attribute domains may skew the distributions of different dimensions, leading to significant perturbations on the published data and thus affecting data utility. Last, dimensionality reduction and noise addition both introduce defection to the published data. How they jointly affect data utility is a tough and open problem.

7.4 Differentially Private Publishing of Dynamic Data

Most of the existing differential privacy research focuses on static data publishing. In practice, many datasets, such as online retail data, recommendation system information and trajectory data, are dynamically updated. Representing dynamic social network data as a static graph and discarding temporal information may result in the loss of evolutionary behaviors of social groups. Thus, how to achieve differential private dynamic social network data publishing is an important research direction.

Differential private publishing of dynamic social network data faces two critical challenges: allocating privacy budget to each data element at each version and handling noise accumulation over continuous data publishing. In an algorithm with multiple sequential queries, the privacy budget may be exhausted after a while based on the notion of composite differential privacy, and thus the promised privacy protection may not be maintained. Therefore, we need a budget allocation strategy that can make the life cycle of privacy budget as long as possible while providing sufficient protection in a composite query. Moreover, since each updated data publishing must consider the added noise in the previous one to counter the correlation between the two releases, the cumulative noise increases rapidly as the number of releases increases, resulting in the fast-decreasing utility in the published data over time.

There exist initial efforts on this direction. For example, Chan *et al.* [110] and Chen *et al.* [111] tackled the continual counting problem and the differential private publishing of sequential data. However, the proposed approaches do not address the failures caused by early exhaustion of privacy budget. The continual release of degree distributions in degree-bounded graphs was considered in [71] but the proposed technique yields poor utility.

Generally speaking, most approaches for differential private data publishing in a static environment cannot be directly applied to publishing of dynamic data. New strategies and mechanisms are highly desirable.

REFERENCES

[1] D. M. Boyd and N. B. Ellison, "Social network sites: Definition, history, and scholarship," *Journal of computer-mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007.

[2] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: an online social network with user-defined privacy," in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, 2009, pp. 135–146.

[3] S. M. Abdulhamid, S. Ahmad, V. O. Waziri, and F. N. Jibril, "Privacy and national security issues in social networks: the challenges," *arXiv preprint arXiv:1402.3301*, 2014.

[4] T. Dalenius, "Towards a methodology for statistical disclosure control," *statistik Tidskrift*, vol. 15, no. 429-444, pp. 2–1, 1977.

[5] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[6] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, pp. 3–es, 2007.

[7] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *2007 IEEE 23rd International Conference on Data Engineering*. IEEE, 2007, pp. 106–115.

[8] R. C.-W. Wong, J. Li, A. W.-C. Fu, and K. Wang, "(α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 754–759.

[9] C. Dwork, "Differential privacy: A survey of results," in *International conference on theory and applications of models of computation*. Springer, 2008, pp. 1–19.

[10] T. Zhu, G. Li, W. Zhou, and S. Y. Philip, "Differentially private data publishing and analysis: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 8, pp. 1619–1638, 2017.

[11] M. U. Hassan, M. H. Rehmani, and J. Chen, "Differential privacy techniques for cyber physical systems: a survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 746–789, 2019.

[12] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks," in *2008 IEEE 24th International Conference on Data Engineering*. IEEE, 2008, pp. 506–515.

[13] E. Zheleva and L. Getoor, "Privacy in social networks: A survey," in *Social network data analytics*. Springer, 2011, pp. 277–306.

[14] S. Sharma, P. Gupta, and V. Bhatnagar, "Anonymisation in social network: A literature survey and classification," *International Journal of Social Network Mining*, vol. 1, no. 1, pp. 51–66, 2012.

[15] J. H. Abawajy, M. I. H. Ninggal, and T. Herawan, "Privacy preserving social network data publication," *IEEE communications surveys & tutorials*, vol. 18, no. 3, pp. 1974–1997, 2016.

[16] S. Ji, P. Mittal, and R. Beyah, "Graph data anonymization, de-anonymization attacks, and de-anonymizability quantification: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1305–1326, 2016.

[17] G. Beigi and H. Liu, "A survey on privacy in social media: Identification, mitigation, and applications," *ACM Transactions on Data Science*, vol. 1, no. 1, pp. 1–38, 2020.

[18] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds. ICALP, Springer, 2006, pp. 1–12.

[19] M. Hay, C. Li, G. Miklau, and D. Jensen, "Accurate estimation of the degree distribution of private networks," in *2009 Ninth IEEE International Conference on Data Mining*. IEEE, 2009, pp. 169–178.

[20] W.-Y. Day, N. Li, and M. Lyu, "Publishing graph degree distribution with node differential privacy," in *Proceedings of the 2016 International Conference on Management of Data*, 2016, pp. 123–138.

[21] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith, "Analyzing graphs with node differential privacy," in *Theory of Cryptography Conference*. Springer, 2013, pp. 457–476.

[22] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, 2007, pp. 75–84.

[23] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Y. Zhao, "Sharing graphs using differentially private graph models," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, 2011, pp. 81–98.

[24] J. Blocki, A. Blum, A. Datta, and O. Sheffet, "Differentially private data analysis of social networks via restricted sensitivity," in *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, 2013, pp. 87–96.

[25] Z. Qin, T. Yu, Y. Yang, I. Khalil, X. Xiao, and K. Ren, "Generating synthetic decentralized social graphs with local differential privacy," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 425–438.

[26] C. Liu, S. Chakraborty, and P. Mittal, "Dependence makes you vulnberable: Differential privacy under dependent tuples." in *NDSS*, vol. 16, 2016, pp. 21–24.

[27] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.

[28] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, 2011, pp. 193–204.

[29] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[30] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2006, pp. 486–503.

[31] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy." *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2014.

[32] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. IEEE, 2007, pp. 94–103.

[33] S. Kotz, T. Kozubowski, and K. Podgorski, *The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance*. Springer Science & Business Media, 2012.

[34] V. Karwa, S. Raskhodnikova, A. Smith, and G. Yaroslavtsev, "Private analysis of graph structure," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 1146–1157, 2011.

[35] F. D. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 2009, pp. 19–30.

[36] B. Yang, I. Sato, and H. Nakagawa, "Bayesian differential privacy on correlated data," in *Proceedings of the 2015 ACM SIGMOD international conference on Management of Data*, 2015, pp. 747–762.

[37] J. Zhao, J. Zhang, and H. V. Poor, "Dependent differential privacy for correlated data," in *2017 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2017, pp. 1–7.

[38] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, 2013, pp. 429–438.

[39] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965.

[40] N. S. Mangat, "An improved randomized response strategy," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 56, no. 1, pp. 93–95, 1994.

[41] R. I. Frederick and H. G. Foster, "Multiple measures of malingering on a forced-choice test of cognitive ability." *Psychological Assessment: A Journal of Consulting and Clinical Psychology*, vol. 3, no. 4, p. 596, 1991.

[42] S. Xiong, A. D. Sarwate, and N. B. Mandayam, "Randomized requantization with local differential privacy," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2189–2193.

[43] A. D. Sarwate and L. Sankar, "A rate-disortion perspective on local differential privacy," in *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2014, pp. 903–908.

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

[45] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," *arXiv preprint arXiv:0903.3276*, 2009.

[46] ——, "Robust de-anonymization of large datasets (how to break anonymity of the netflix prize dataset)," *University of Texas at Austin*, 2008.

[47] R. Dey, C. Tang, K. Ross, and N. Saxena, "Estimating age privacy leakage in online social networks," in *2012 proceedings ieee infocom*. IEEE, 2012, pp. 2836–2840.

[48] N. Z. Gong, A. Talwalkar, L. Mackey, L. Huang, E. C. R. Shin, E. Stefanov, E. Shi, and D. Song, "Joint link prediction and attribute inference using a social-attribute network," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 2, pp. 1–20, 2014.

[49] S. H. Ahmadinejad and P. W. Fong, "Unintended disclosure of information: Inference attacks by third-party extensions to social network systems," *Computers & security*, vol. 44, pp. 75–91, 2014.

[50] S. Labitzke, F. Werling, J. Mittag, and H. Hartenstein, "Do online social network friends still threaten my privacy?" in *Proceedings of the third ACM conference on Data and application security and privacy*, 2013, pp. 13–24.

[51] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel, "You are who you know: inferring user profiles in online social

networks," in *Proceedings of the third ACM international conference on Web search and data mining*, 2010, pp. 251–260.

[52] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft, "Blurme: Inferring and obfuscating user gender based on ratings," in *Proceedings of the sixth ACM conference on Recommender systems*, 2012, pp. 195–202.

[53] A. Chaabane, G. Acs, M. A. Kaafar *et al.*, "You are what you like! information leakage through users' interests," in *Proceedings of the 19th Annual Network & Distributed System Security Symposium (NDSS)*. Citeseer, 2012.

[54] M. Kosinski, D. Stillwell, and T. Graepel, "Private traits and attributes are predictable from digital records of human behavior," *Proceedings of the national academy of sciences*, vol. 110, no. 15, pp. 5802–5805, 2013.

[55] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel, "A practical attack to de-anonymize social network users," in *2010 IEEE Symposium on Security and Privacy*. IEEE, 2010, pp. 223–238.

[56] S. Ji, W. Li, N. Z. Gong, P. Mittal, and R. A. Beyah, "On your social network de-anonymizablity: Quantification and large scale evaluation with seed knowledge." in *NDSS*, 2015.

[57] S. Ji, W. Li, M. Srivatsa, and R. Beyah, "Structural data de-anonymization: Quantification, practice, and implications," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 1040–1053.

[58] J. Qian, X.-Y. Li, C. Zhang, and L. Chen, "De-anonymizing social networks and inferring private attributes using knowledge graphs," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

[59] S. Ji, T. Wang, J. Chen, W. Li, P. Mittal, and R. Beyah, "De-sag: On the de-anonymization of structure-attribute graph data," *IEEE Transactions on Dependable and Secure Computing*, 2017.

[60] F. Shirani, S. Garg, and E. Erkip, "Optimal active social network de-anonymization using information thresholds," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 1445–1449.

[61] Y. Shao, J. Liu, S. Shi, Y. Zhang, and B. Cui, "Fast de-anonymization of social networks with structural information," *Data Science and Engineering*, vol. 4, no. 1, pp. 76–92, 2019.

[62] M. Kiranmayi and N. Maheswari, "A review on privacy preservation of social networks using graphs," *Journal of Applied Security Research*, pp. 1–34, 2020.

[63] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 181–190.

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

[65] K. Liu and E. Terzi, "Towards identity anonymization on graphs," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 93–106.

[66] G. Kossinets and D. J. Watts, "Empirical analysis of an evolving social network," *science*, vol. 311, no. 5757, pp. 88–90, 2006.

[67] C. Task and C. Clifton, "A guide to differential privacy theory in social network analysis," in *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 2012, pp. 411–417.

[68] ——, "What should we protect? defining differential privacy for social network analysis," in *State of the Art Applications of Social Network Analysis*. Springer, 2014, pp. 139–161.

[69] M. Hay, V. Rastogi, G. Miklau, and D. Suciu, "Boosting the accuracy of differentially private histograms through consistency," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 1021–1032, 2010.

[70] K. R. Macwan and S. J. Patel, "Node differential privacy in social graph degree publishing," *Procedia computer science*, vol. 143, pp. 786–793, 2018.

[71] S. Song, S. Little, S. Mehta, S. Vinterbo, and K. Chaudhuri, "Differentially private continual release of graph statistics," *arXiv preprint arXiv:1809.02575*, 2018.

[72] S. Raskhodnikova and A. Smith, "Efficient lipschitz extensions for high-dimensional graph statistics and node private degree distributions," *arXiv preprint arXiv:1504.07912*, 2015.

[73] J. Ullman and A. Sealfon, "Efficiently estimating erdos-renyi graphs with node differential privacy," in *Advances in Neural Information Processing Systems*, 2019, pp. 3765–3775.

[74] F. Ahmed, A. X. Liu, and R. Jin, "Publishing social network graph eigen-spectrum with privacy guarantees," *IEEE Transactions on Network Science and Engineering*, 2019.

[75] A. Gupta, A. Roth, and J. Ullman, "Iterative constructions and private data release," in *Theory of cryptography conference*. Springer, 2012, pp. 339–356.

[76] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "Private release of graph statistics using ladder functions," in *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, 2015, pp. 731–745.

[77] V. Rastogi, M. Hay, G. Miklau, and D. Suciu, "Relationship privacy: output perturbation for queries with joins," in *Proceedings of the twenty-eighth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2009, pp. 107–116.

[78] J. Gehrke, E. Lui, and R. Pass, "Towards privacy for social networks: A zero-knowledge based definition of privacy," in *Theory of cryptography conference*. Springer, 2011, pp. 432–449.

[79] L. Liu, J. Wang, J. Liu, and J. Zhang, "Privacy preserving in social networks against sensitive edge disclosure," Technical Report CMIDA-HiPSCCS 006-08, Tech. Rep., 2008.

[80] S. Das, Ö. Eğecioğlu, and A. El Abbadi, "Anonymizing weighted social network graphs," in *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*. IEEE, 2010, pp. 904–907.

[81] S. Costea, M. Barbu, and R. Rughinis, "Qualitative analysis of differential privacy applied over graph structures," in *2013 11th RoEduNet International Conference*. IEEE, 2013, pp. 1–4.

[82] X. Li, J. Yang, Z. Sun, and J. Zhang, "Differential privacy for edge weights in social networks," *Security and Communication Networks*, vol. 2017, 2017.

[83] R. Bassily and A. Smith, "Local, private, efficient protocols for succinct histograms," in *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, 2015, pp. 127–135.

[84] P. Kairouz, K. Bonawitz, and D. Ramage, "Discrete distribution estimation under local privacy," *arXiv preprint arXiv:1602.07387*, 2016.

[85] R. Chen, H. Li, A. K. Qin, S. P. Kasiviswanathan, and H. Jin, "Private spatial data aggregation in the local setting," in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 2016, pp. 289–300.

[86] S. Wang, L. Huang, P. Wang, Y. Nie, H. Xu, W. Yang, X.-Y. Li, and C. Qiao, "Mutual information optimally local private discrete distribution estimation," *arXiv preprint arXiv:1607.08025*, 2016.

[87] M. Ye and A. Barg, "Optimal schemes for discrete distribution estimation under locally differential privacy," *IEEE Transactions on Information Theory*, vol. 64, no. 8, pp. 5662–5676, 2018.

[88] T. T. Nguyên, X. Xiao, Y. Yang, S. C. Hui, H. Shin, and J. Shin, "Collecting and analyzing data from smart device users with local differential privacy," *arXiv preprint arXiv:1606.05053*, 2016.

[89] G. Fanti, V. Pihur, and Ú. Erlingsson, "Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 3, pp. 41–61, 2016.

[90] X. Ren, C.-M. Yu, W. Yu, S. Yang, X. Yang, J. A. McCann, and S. Y. Philip, "Lopub: High-dimensional crowdsourced data publication with local differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2151–2166, 2018.

[91] M. J. Wainwright, M. I. Jordan, and J. C. Duchi, "Privacy aware learning," in *Advances in Neural Information Processing Systems*, 2012, pp. 1430–1438.

[92] A. Cheu, A. Smith, J. Ullman, D. Zeber, and M. Zhilyaev, "Distributed differential privacy via shuffling," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2019, pp. 375–403.

[93] H. Sun, X. Xiao, I. Khalil, Y. Yang, Z. Qin, H. Wang, and T. Yu, "Analyzing subgraph statistics from extended local views with decentralized differential privacy," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 703–717.

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

[95] W. Aiello, F. Chung, and L. Lu, "A random graph model for massive graphs," in *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, 2000, pp. 171–180.

[96] Q. Ye, H. Hu, M. H. Au, X. Meng, and X. Xiao, "Towards locally

differentially private generic graph metric estimation," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 2020, pp. 1922–1925.

[97] N. Li, W. Qardaji, and D. Su, "On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, 2012, pp. 32–33.

[98] E. Candès and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, no. 2, pp. 21 – 30, March 2008.

[99] B. Zhang, X. Cheng, N. Zhang, Y. Cui, Y. Li, and Q. Liang, "Sparse target counting and localization in sensor networks based on compressive sensing," in *The 30th IEEE Conference on Computer Communications (IEEE INFOCOM 2011)*, Shanghai, China, April 2011, pp. 2255–2263.

[100] Y. Tang, B. Zhang, T. Jing, D. Wu, and X. Cheng, "Robust compressive data gathering in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 6, pp. 2754–2761, June 2013.

[101] C. Zhang, H. Jiang, X. Cheng, F. Zhao, Z. Cai, and Z. Tian, "Utility analysis on privacy-preservation algorithms for online social networks: An empirical study," *Pervasive and Ubiquitous Computing*, August 2019.

[102] J. Hsu, M. Gaboardi, A. Haeberlen, S. Khanna, A. Narayan, B. C. Pierce, and A. Roth, "Differential privacy: An economic method for choosing epsilon," in *2014 IEEE 27th Computer Security Foundations Symposium*. IEEE, 2014, pp. 398–410.

[103] L. Cui, Y. Qu, M. R. Nosouhi, S. Yu, J.-W. Niu, and G. Xie, "Improving data utility through game theory in personalized differential privacy," *Journal of Computer Science and Technology*, vol. 34, no. 2, pp. 272–286, 2019.

[104] A. R. Chowdhury, T. Rekatsinas, and S. Jha, "Data-dependent differentially private parameter learning for directed graphical models," *arXiv preprint arXiv:1905.12813*, 2019.

[105] C. Dwork, N. Kohli, and D. Mulligan, "Differential privacy in practice: Expose your epsilons!" *Journal of Privacy and Confidentiality*, vol. 9, no. 2, 2019.

[106] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "Privbayes: Private data release via bayesian networks," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 4, pp. 1–41, 2017.

[107] C. Xu, J. Ren, Y. Zhang, Z. Qin, and K. Ren, "Dppro: Differentially private high-dimensional data release via random projection," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 3081–3093, 2017.

[108] H. Li, L. Xiong, and X. Jiang, "Differentially private synthesization of multi-dimensional data using copula functions," in *Advances in database technology: proceedings. International conference on extending database technology*, vol. 2014. NIH Public Access, 2014, p. 475.

[109] R. Chen, Q. Xiao, Y. Zhang, and J. Xu, "Differentially private high-dimensional data publication via sampling-based inference," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 129–138.

[110] T.-H. H. Chan, E. Shi, and D. Song, "Private and continual release of statistics," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 3, pp. 1–24, 2011.

[111] R. Chen, G. Acs, and C. Castelluccia, "Differentially private sequential data publication via variable-length n-grams," in *Proceedings of the 2012 ACM conference on Computer and communications security*, 2012, pp. 638–649.

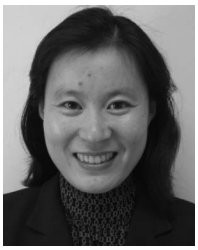
Honglu Jiang received her M.S. degree from Qufu Normal University in 2012, and Ph.D. degree from The George Washington University in 2021. Her research interests include wireless networks, differential privacy, big data and privacy preservation.

Jian Pei 's professional interest is to facilitate efficient, fair, and sustainable usage of data and data analytics for social, commercial and ecological good. Through inventing, implementing and deploying a series of data mining principles and methods, he produced remarkable values to academia and industry. His algorithms have been adopted by industry, open source toolkits and textbooks. His publications have been cited more than 98,000 times. He is also an active and productive volunteer for professional community services, such as chairing ACM SIGKDD, running many premier academic conferences in his areas, and being editor-in-chief or associate editor for the flagship journals in his fields. He is recognized as a fellow of the Royal Society of Canada (i.e., the national academy of Canada), a fellow of the Canadian Academy of Engineering, a fellow of ACM, and a fellow of IEEE. He received a series of prestigious awards, such as the ACM SIGKDD Innovation Award, the ACM SIGKDD Service Award, and the IEEE ICDM Research Award. Currently he is a full professor at Simon Fraser University, Canada.

Dongxiao Yu received the BS degree from the School of Mathematics, Shandong University, in 2006 and the PhD degree from the Department of Computer Science, The University of Hong Kong, in 2014. He is currently a professor with the School of Computer Science and Technology, Shandong University. His research interests include wireless networks and distributed computing. He is a member of the IEEE.

Jiguo Yu received his Ph.D. degree in School of mathematics from Shandong University in 2004. He became a full professor in the School of Computer Science, Qufu Normal University, Shandong, China in 2007. Currently he is a full professor in Qilu University of Technology (Shandong Academy of Sciences). His main research interests include privacy-aware computing, wireless networking, distributed algorithms, peer-to-peer computing, and graph theory. Particularly, he is interested in designing and analyzing algorithms for many computationally hard problems in networks. He is a senior member of IEEE, a member of ACM and a senior member of the CCF (China Computer Federation).

Bei Gong received his BS degree from Shandong University in 2005, and PhD degree from Beijing University of Technology in 2012. He has six National invention patents and one monograph textbook. His research interests include trusted computing, Internet of things security, mobile Internet of things, and mobile edge computing. He is the principle investigator of 8 national projects such as the National Natural Science Foundation grants and 6 provincial and ministerial projects such as the general science and technology program of Beijing Municipal Education Commission.



Xiuzhen Cheng received her M.S. and Ph.D. degrees in computer science from the University of Minnesota—Twin Cities in 2000 and 2002, respectively. She is a professor of Computer Science at Shandong University, P. R. China. Her current research focuses on Blockchain computing, privacy-aware computing, and wireless and mobile security. She served/is serving on the editorial boards of several technical journals and the technical program committees of various professional conferences/workshops. She was a faculty member in the Department of Computer Science at The George Washington University from September 2002 to August 2020, and worked as a program director for the US National Science Foundation (NSF) from April to October in 2006 (full time) and from April 2008 to May 2010 (part time). She received the NSF CAREER Award in 2004. She is a member of ACM, and a Fellow of IEEE.