Group Privacy-aware Disclosure of Association Graph Data

Balaji Palanisamy
School of Computing and Information
University of Pittsburgh
Pittsburgh, USA
Email: bpalan@pitt.edu

Chao Li
School of Computing and Information
University of Pittsburgh
Pittsburgh, USA
Email: chl205@pitt.edu

Prashant Krishnamurthy
School of Computing and Information
University of Pittsburgh
Pittsburgh, USA
Email: prashk@pitt.edu

***Abstract*—In the age of Big Data, we are witnessing a huge proliferation of digital data capturing our lives and our surroundings. Data privacy is a critical barrier to data analytics and privacy-preserving data disclosure becomes a key aspect to leveraging large-scale data analytics due to serious privacy risks. Traditional privacy-preserving data publishing solutions have focused on protecting individual's private information while considering all aggregate information about individuals as safe for disclosure. This paper presents a new privacy-aware data disclosure scheme that considers group privacy requirements of individuals in bipartite association graph datasets (e.g., graphs that represent associations between entities such as customers and products bought from a pharmacy store) where even aggregate information about groups of individuals may be sensitive and need protection. We propose the notion of** ϵ_g*-Group Differential Privacy* **that protects sensitive information of groups of individuals at various defined group protection levels, enabling data users to obtain the level of information entitled to them. Based on the notion of group privacy, we develop a suite of differentially private mechanisms that protect group privacy in bipartite association graphs at different group privacy levels based on specialization hierarchies. We evaluate our proposed techniques through extensive experiments on three real-world association graph datasets and our results demonstrate that the proposed techniques are effective, efficient and provide the required guarantees on group privacy.**

I. INTRODUCTION

In the age of Big Data, organizations and governments can obtain rich information and insights by mining large volumes of data that are generated at an unprecedented velocity, volume and scale[3], [6], [12]. Data privacy becomes a critical barrier in effectively leveraging large-scale data analytics due to serious privacy risks[1], [30]. Publishing and maintaining data that contains sensitive information about individuals is a challenging problem. Such sensitive datasets may include private information such as medical information, patient records, census information or sales transactions made by customers. Private data also arise in the form of associations between entities in real world such as the drugs purchased by patients in a pharmacy store or the movies rated by viewers in a movie rating database or the communication between friends in an online social network[4], [8]. In general, the associations between the entities (such as the drugs purchased by an individual patient or the movies rated by an individual viewer) are considered sensitive and such associations are naturally represented as large, sparse bipartite graphs[8] with nodes representing the entities (e.g., drugs and patients) and the edges representing the associations between them (e.g., purchases of the drugs made by the patients).

Publishing real-world association data in a privacy-conscious manner is critical for a number of purposes. For instance, medical scientists may want to study the outbreaks of new diseases based on the type of drugs administered to patients and drug manufacturers may wish to perform business analytics based on the purchase trends of the drugs. In the past, data privacy schemes [7], [10], [13], [17], [18], [19], [20], [25], [26], [27] have primarily focused on protecting individuals' information in sensitive datasets while allowing aggregate information on groups of individuals. Differential privacy [10] provides a model to quantify the disclosure risks by ensuring that the published statistical data does not depend on the presence or absence of a single individual's record in the dataset[10], [11]. These schemes developed with an intrinsic assumption that all aggregate information in a dataset is safe for disclosure do not consider the scenarios when some aggregate information itself can be sensitive. Sensitive information may arise either as: (i) an individual sensitive value indicating an individual's private information (e.g., did buyer 'Bob' purchase the drug 'insulin'?) in a dataset or (ii) a statistical value representing some sensitive statistics about a group/sub-group of individuals (e.g., the total number of 'Psychiatric' drugs purchased by buyers in a given neighborhood represented by a zipcode). Such group privacy requirements may also result in multi-level privacy controlled situations where data users may have different levels of access to the data at different privacy levels. For example, in a drug purchase association graph dataset, we may have a need to protect group privacy at different protection levels based on the access privilege of the data users. Some data users (e.g., less privileged data analysts) may be allowed to obtain graph structure and aggregate information for a larger group size (e.g., the number of purchases of 'Psychiatric' drugs in the city of 'Pittsburgh') while some other more privileged data users may have access to the same information at smaller group sizes (e.g., the number of 'Psychiatric' drugs purchased in the zipcode '15206' within 'Pittsburgh'). While existing mechanisms [7], [10], [13], [19], [20], [26], [27], [31] have focused on protecting individual's sensitive values in datasets, this paper proposes a privacy-preserving data publishing mechanism addressing group privacy when aggregate information about groups of individuals can be sensitive and needs protection.

PID	DOB	Sex	Zipcode
P1	7/18/87	F	19130
P2	2/17/83	M	90031
P3	5/07/77	M	94107
P4	1/5/76	F	19181
P5	8/4/82	M	94177
P6	3/9/79	M	90101
P7	4/10/64	M	15203
P8	2/6/81	F	15217

TABLE I: Patients

DID	Drug name	Sub category	Main category
D1	CItalopram	SSRIs	Antidepressants
D2	Phenelzine	MAOIs	Antidepressants
D3	Erythromycin	Macrolide	Antibiotic
D4	Selegiline	MAOIs	Antidepressants
D5	Azithromycin	Macrolide	Antibiotic
D6	Cephalosporin	Beta-Lactams	Antibiotic
D7	Penicillines	Beta-Lactams	Antibiotic
D8	Fluoxetine	SSRIs	Antidepressants

TABLE II: Drugs

PID	DID
P1	D6
P2	D1
P3	D4
P3	D7
P4	D6
P5	D8
P6	D2
P7	D3
P7	D8
P8	D5
P8	D7

TABLE III: Associations

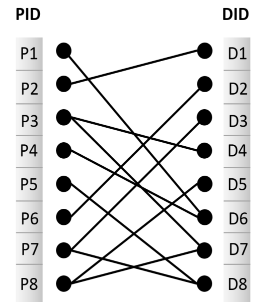


Fig. 1: Graph

Concretely, this paper makes the following key contributions: We first propose the notion of ϵ_g-group differential privacy that provides guaranteed protection of aggregate information of a group of individuals in a given dataset. Second, based on the notion of ϵ_g-group differential privacy, we study the group privacy problem (Section 2) in the context of bipartite associations graphs and develop a suite of differentially private mechanisms (Section 3) that guarantee group privacy at variously defined group granularity levels. We show that this model can be used to support multi-level privacy that provides different levels of group granularity to users based on access privileges. Finally, we evaluate the proposed techniques through extensive experiments on three real-world association graph datasets (Section 4) and our results demonstrate that the proposed techniques are effective, efficient and provide the required guarantees on group privacy.

II. Concepts and Models

In this section, we review the fundamental concepts related to association graphs and define the group privacy-aware multi-level privacy protection problem. We also review the conventional differential privacy model for protecting individual privacy and present the proposed notion of ϵ_g-group differential privacy.

A. Bipartite Association Graphs

We represent a bipartite graph as $BG = (V, W, E)$. The graph BG consists of $m = |V|$ nodes of a first type and $n = |W|$ of a second type and a set of edges $E \subseteq V \times W$.Thus, a bipartite graph can be simply represented as a set of two-node pairings, where a two-node pairing (a, b) represents an edge in E between node $a \in V$ and node $b \in W$. For instance, the bipartite graph could represent the associations of patients and drugs based on the purchases made by them or the movies watched by individual viewers in a movie rating database. In that case, the set of nodes, V represents patients and W represents drugs and any edge (p, d) in E will represent the association that the patient p bought the drug d. We note that these association graphs are quite sparse. Each patient buys only a very small subset of the set of all available drugs. Similarly, each viewer watches and rates only a small subset of all available movies. We show an example of such a bipartite graph in Figure 1 where the nodes represent drugs and patients and the edges represent the drugs purchased by the patients. The details of the patients and drugs are shown in Tables I - II and the associations are shown in Table III.

B. Group Privacy and Multi-level protection

Sensitive information in an association graph may arise either as: (i) an individual sensitive value indicating an individual's private information (e.g., did buyer 'Bob' purchase the drug 'insulin'?) or (ii) a statistical value representing some sensitive statistics about a group/sub-group of individuals (e.g., the total number of 'Psychiatric' drug purchases made by buyers in a given neighborhood represented by a zipcode). While existing mechanisms[7], [10], [13], [19], [20], [26], [27], [31] have focused on protecting individual's sensitive values, this paper proposes a privacy-preserving data publishing mechanism addressing group privacy concerns when aggregate information about groups of individuals is sensitive and needs protection. In a drug purchase association graph, one may need to protect group privacy at different protection levels depending on the access privilege of the data users. For instance, in the example shown in Figure 3, some data users (e.g., less privileged data analysts) may be allowed to access the published graph at access level, L_2. Such a user can infer the structural properties and aggregate information at course granular groups (e.g., the total number of antidepressants purchased by California residents). Some other higher privileged data users may be allowed to access the graph at access level, L_1 in which he/she may obtain information about fine-grained groups in the graph (e.g., how many residents in San Francisco purchased a SSRIs type antidepressant?).

In general, the queries in a bipartite association graph may use the graph structure characteristics in addition to attribute predicates (e.g., in the drug purchase dataset, the number customers in the Zipcode 30323 who had purchased 3 or more different kinds of antibiotic drugs will require structural characteristics of the graph for processing). Thus, the group privacy-aware graph perturbation process should retain as many structural properties as possible after the perturbation process. We next introduce the notion of conventional differential privacy that protects the inference of a single individual's record in a dataset.

C. Differential Privacy

Differential privacy is a classical privacy definition [10] that makes conservative assumptions about the adversary's background knowledge and protects a single individual's privacy by considering adjacent data sets which differ only in one record. Formally, a data set D can be considered as a subset of records from the universe U, represented by $D \in \mathbb{N}^{|U|}$, where $\mathbb{N}$ stands for the non-negative set and D_i is the number of element i in $\mathbb{N}$. For example, in the case of a bipartite graph, if $U = \{(a, c), (a, d), (b, d)\}$, $D = \{(a, c), (a, d), (b, d)\}$ can be represented as $\{1, 1, 1\}$ as it contains each element of U once. Similarly, $D' = \{(a, c), (b, d)\}$ can be represented as $\{1, 0, 1\}$ as it does not contain $\{(a, d)\}$. Based on this representation, it is appropriate to use l_1 distance (Manhattan distance) to measure the distance between data sets.

Definition *1 (*Data set Distance*): The l_1 distance between two data sets D_1 and D_2 is defined as $||D_1 - D_2||_1$,* *which is calculated by:*

$$||D_1 - D_2||_1 = \sum_{i=1}^{|U|} |D_{1_i} - D_{2_i}|$$

The manhattan distance between the datasets leads us the notion of adjacent data sets.

DEFINITION *2 (*ADJACENT DATA SET*): Two data sets D_1, D_2 are adjacent data sets of each other if $||D_1 - D_2||_1 = 1$.*

Based on the notion of adjacent datasets defined above, differential privacy can be defined formally as follows.

DEFINITION *3 (*DIFFERENTIAL PRIVACY*): A randomized algorithm $\mathcal{A}$ guarantees (ϵ, δ)-differential privacy if for all adjacent data sets D_1 and D_2 differing by at most one record, and for all possible results $\mathcal{S} \subseteq Range(\mathcal{A})$,*

$$Pr[\mathcal{A}(D_1) = \mathcal{S}] \leq e^{\epsilon} \times Pr[\mathcal{A}(D_2) = \mathcal{S}] + \delta$$

where the probability space is over the randomness of $\mathcal{A}$.

Differential privacy ensures that even when the adversary knows all the records in a data set D except that of a target individual, the probability of inferring that information is restricted by an upper bound.

D. Group Differential Privacy

In this work, we extend the conventional notion of differential privacy model to protect group privacy at various group granularity levels. We focus on the scenarios where one needs to protect group-level privacy in addition to individual privacy, where a group consists of a set of individuals. We define the proposed notion of ϵ_g - group differential privacy by considering adjacent data sets from a group privacy perspective. Figure 2(a) shows an example dataset of patients (with patient IDs, PID) belonging to different zipcodes. In Figure 2(b), we partition the universe, U into N non-overlapping subgroups, $U = \cup_{i=1}^{n} G_i$, $G = \{G_1, ..., G_n\}$ with each record of U joining only one subgroup $G_i \in G$. Here N represents the natural number set. Therefore, the overall data set space can be represented as $D = \{D_i | D_i = \cup_{i \in I} G_i, G_i \in G, I \subseteq \{1, ..., N\}\}$ as shown in Figure 2(c). This leads to a number of group-level adjacent data sets as shown in Figure 2(d). Formally, group-level adjacent data sets are defined as

DEFINITION *4 (*GROUP-LEVEL ADJACENT DATA SETS*): Two data sets D_1 and D_2 are group-level adjacent data sets of each other if $\exists G_i \in G$ such that $D_1 = D_2 \cup G_i$.*

Thus the notion of ϵ_g- group differential privacy based on level adjacent datasets is defined as

DEFINITION *5 (*GROUP DIFFERENTIAL PRIVACY*): A randomized algorithm $\mathcal{A}$ guarantees ϵ_g- group differential privacy if for all adjacent data sets D_1 and D_2 differing by at most one group, $G_i \in G$, and for all possible results $\mathcal{S} \subseteq Range(\mathcal{A})$,*

$$Pr[\mathcal{A}(D_1) = \mathcal{S}] \leq e^{\epsilon_g} \times Pr[\mathcal{A}(D_2) = \mathcal{S}]$$

where the probability space is over the randomness of $\mathcal{A}$.

E. Differential Privacy Mechanisms

Many randomized algorithms have been proposed to guarantee differential privacy. We briefly introduce the most commonly used differentially private mechanisms namely the Laplace Mechanism[10], the Gaussian Mechanism[11] and the Exponential Mechanism[22].

Laplace Mechanism: Given a data set D, a function f and the budget ϵ, the Laplace Mechanism first calculates the actual $f(D)$ and then perturbs this true answer by adding a noise[10]. The noise is calculated based on a Laplace random variable, with the variance $\lambda = \triangle f / \epsilon$, where $\triangle f$ is the l_1 sensitivity.

DEFINITION *6 (l_1* SENSITIVITY *[11]): Given a function $f : \mathbb{N}^{|U|} \rightarrow \mathbb{R}^d$, the l_1 sensitivity is measured as:*

$$\triangle f = \max_{\substack{D_1, D_2 \in \mathbb{N}^{|U|} \\ ||D_1 - D_2||_1 = 1}} ||f(D_1) - f(D_2)||_1$$

where $||f(D_1) - f(D_2)||_1 = |f(D_1) - f(D_2)|$ is the Manhattan Distance.

In other words, l_1 sensitivity measures the maximum impact that can be caused by changing a single record. It is only related to the function f itself, but independent of the data sets.

DEFINITION *7 (*LAPLACE MECHANISM *[10]): Given a function $f : \mathbb{N}^{|U|} \rightarrow \mathbb{R}^d$, a budget ϵ and a data set D, for each output,*

$$\mathcal{A}_{LM}(D, f, \epsilon) = f(D) + Lap(\triangle f / \epsilon)$$

where $Lap(\triangle f / \epsilon)$ is a random variable sampled from the Laplace distribution with 0 mean and $\triangle f / \epsilon$ variance.

Gaussian Mechanism: Instead of adding Laplace noise to achieve $(\epsilon, 0)$-differential privacy, it is possible to achieve (ϵ, δ)-differential privacy using a Gaussian noise[11]. When δ is small, the gap between the two privacy level is small.

DEFINITION *8 (l_2* SENSITIVITY *[11]): Given a function $f : \mathbb{N}^{|U|} \rightarrow \mathbb{R}$, the l_2 sensitivity is measured as:*

$$\triangle_2 f = \max_{\substack{D_1, D_2 \in \mathbb{N}^{|U|} \\ ||D_1 - D_2||_1 = 1}} ||f(D_1) - f(D_2)||_2$$

where $||f(D_1) - f(D_2)||_2 = \sqrt{|f(D_1) - f(D_2)|^2}$ is the Euclidean Distance.

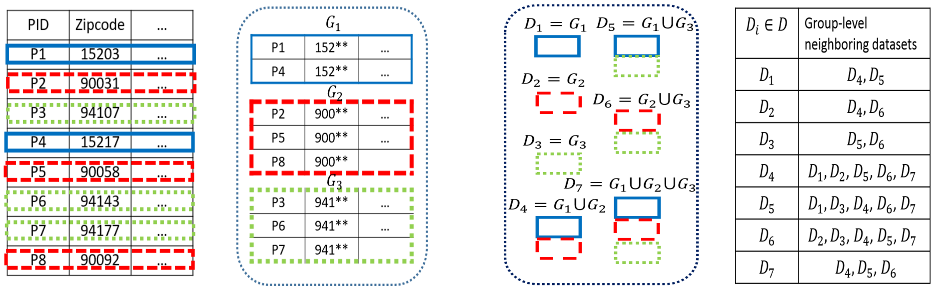
For real-valued functions, $||f(D_1) - f(D_2)||_1 = ||f(D_1) - f(D_2)||_2$, so $\triangle f = \triangle_1 f = \triangle_2 f$.

DEFINITION *9 (*GAUSSIAN MECHANISM *[11]): Given a function $f : \mathbb{N}^{|U|} \rightarrow \mathbb{R}$, a budget $\epsilon \in (0, 1)$, a δ and a data set D, for each output,*

$$\mathcal{A}_{GM}(D, f, \epsilon) = f(D) + Gaus(c\triangle_2 f / \epsilon)$$

where $Gaus(c\triangle_2 f / \epsilon)$ is a random variable sampled from the Gaussian distribution with 0 mean and $c\triangle_2 f / \epsilon$ variance, and $c^2 > 2ln(1.25/\delta)$

Exponential Mechanism: Unlike Laplace Mechanism and Gaussian Mechanism, the Exponential Mechanism is proposed to give differential privacy for non-numerical data sets[22]. Given an output range $\mathbb{R}$, a utility function $u : (D \times \mathcal{R}) \rightarrow \mathbb{R}$



(a) Universe U (b) Group space G (c) Data set space D (d) Group adjacent data sets

Fig. 2: Group-level adjacent data sets

is designed to assign a score for each $r \in \mathbb{R}$, where higher scores means higher utility and is expected to be given higher probability to be chosen. Therefore, the exponential mechanism builds a probability distribution over the whole range $\mathbb{R}$ and takes one sample as the output. The sensitivity of utility function u is

$$\triangle u = \max_{\substack{D_1, D_2 \in \mathbb{N}^{|U|} \\ ||D_1 - D_2||_1 = 1}} |u(D_1, r) - u(D_2, r)|$$

DEFINITION *10 (*EXPONENTIAL MECHANISM *[22]): Given a budget ϵ, a data set D, an output range $\mathcal{R}$ and a utility function $u : (D \times \mathcal{R}) \rightarrow \mathbb{R}$,* the Exponential Mechanism $\mathcal{A}_{EM}$ selects and outputs each $r \in \mathbb{R}$ with probability proportional to $\exp(\frac{\epsilon u(D,r)}{2 \triangle u})$.

In the next section, we employ these differential privacy mechanisms to develop group differential privacy aware mechanisms for disclosure of association graphs.

III. GROUP PRIVACY-AWARE DISCLOSURE

We present our proposed techniques for supporting group-privacy aware disclosure bipartite association graphs considering the guarantees of group privacy requirements in a dataset. Our proposed approach consists of two parts: (i) the first part of the proposed approach, namely *DiffPar* hierarchically partitions and groups the nodes and edges of the given association graph into different levels of granularity of disclosure in terms of group size considering the sensitivity of the formed groups and (ii) the second component of the algorithm, namely *DiffAggre* performs a bottom-up aggregation and noise injection to guarantee ϵ_g-group differential privacy in the published dataset. In *DiffPar*, the groups on the left and right sides of a bipartite graph are specialized iteratively and partitioned into a set of fine granular smaller sub groups. Each left subgroup is connected to one or more right subgroups through associations (edges), forming a subgraph. Therefore, after n specializations, the raw graph is partitioned up to 4^n subgraphs. *DiffPar* employs an exponential mechanism[22] to ensure that the partitioning process is differentially private. After the input graph is specialized and partitioned using *DiffPar*, *DiffAggre* injects carefully calibrated Gaussian noise to ensure group differential privacy of each group at a given privacy level. The proposed approach protects against the inference of the number of edges between the sub groups (within the subgraphs) through the injection of random noise while retaining the structural properties of the subgraphs even after the addition of the random noise. We illustrate these algorithms in detail in the following subsections.

A. *Top-down Group Partitioning*

The objective of the *DiffPar* partitioning algorithm is to partition the nodes of the bipartite graph through a series of specializations such that the sensitivity for each level in the classification hierarchy is minimized. In other words, the algorithm tries to reduce the noise required to be injected for guaranteeing group differential privacy. An example of the partitioning process of *DiffPar* is shown in Figure 3 where a three level classification is obtained as a result of two specializations. The level L_3 in the figure indicates the raw input association graph with node IDs namely 'Patient ID' and 'Drug ID' and with attributes namely 'Zipcode' and 'Drug name'. Each specialization of this raw graph splits both the left group represented by 'All Zipcodes' and the right group represented by'All drugs' to two sub-groups and creates $2X2$ subgraphs for level L_2. The subgroups are represented by 'Pennsylvania-Antidepressants', 'Pennsylvania-Antibiotic', 'California-Antidepressants' and 'California-Antibiotic' respectively. By performing another specialization for each of the four subgraphs at level L_2, we can generate $4X4$ fine-grained subgraphs for level L_1. In this way, subgraphs at different levels can be disclosed to users with different privileges. Typically, with higher privilege, users can obtain more fine-grained subgraphs. However, before subgraphs at a certain level, say L_a, are released, noises are injected to protect group differential privacy for a lower level, say L_b where $b < a$ such that the disclosed data guarantees the required group privacy. We denote such group-differentially private bipartite graph disclosure as $L'_{a(b)}$ that represents that subgraphs at level a are disclosed with group differential privacy protection for fine-grained subgraphs at level b in the disclosed data. We refer to L_a as the disclosure level and L_b as the protection level. For example, $L'_{2(1)}$ in Figure 3 denotes that in the disclosed data, data users can view the 4 subgraphs at disclosure level L_2 while group differential privacy of the 16 fine-grained subgraphs at protection level L_1 is protected.

In order to reduce the required noise under a fixed budget to protect group differential privacy, the sensitivity needs to be minimized during the specialization and splitting process. For example, if a user is allowed to access the count of edges within each subgraph at level L_3 with group differential privacy protected for level L_2, the sensitivity in this context refers to the maximum contribution by a single subgraph at level L_2. If the entire bipartite graph contains 20000 edges, theoretically the contribution (in terms of edge count) of level L_2 subgraph can be any value in the range $[0, 20000]$. Therefore, the maximum influence caused by changing one level

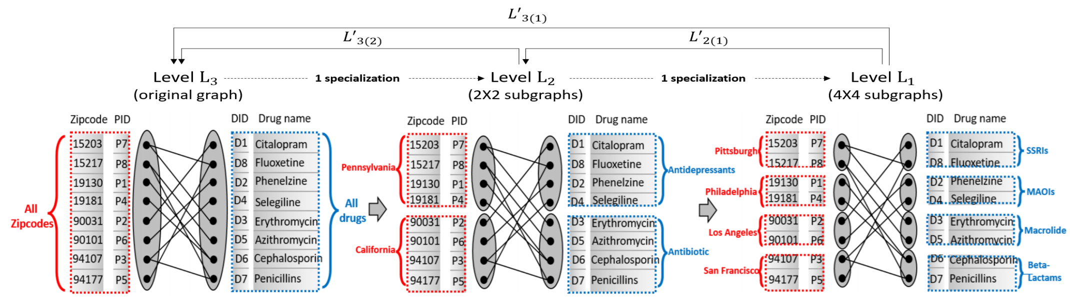


Fig. 3: top-down group partitioning

L_2 subgraph is the maximum value within this range, namely 20000. Such a high sensitivity can generate an unacceptably higher noise to guarantee ϵ_g-*group differential privacy*, making the final published data less useful. A key objective of the partitioning is to provide an upper bound on the maximum number of edges that can be contained in a subgraph at a given hierarchy level (eg., level L_2). Given that association graphs are extremely sparse (i.e., a single buyer buys only a few drugs among the list of all available drugs in a pharmacy store, a single viewer watches only a very small subset of all movies available in a movie database), we note that it is possible to drastically reduce the sensitivity of the partitioned graph using an appropriate differentially private node grouping. Inspired by this observation, the proposed *DiffPar* aims to determine the most appropriate split points in the specialization to intelligently partition the bipartite graph to minimize the group-level sensitivity while operating the algorithm under a differential privacy budget. *DiffPar* achieves the minimized sensitivity through a three-step process. First, after one specialization in each subgroup at level L_i ($i \in [2, n]$ where n is the total number of levels), the number of edges E within this subgraph is divided into four parts, namely $E1, E2, E3, E4$, where $E1+E2+E3+E4 = E$. For example, in Figure 3, the total number of edges $E = 11$ between the left group $All\ Zipcode$ and right group $All\ drug$ at level L_3 is composed of $E_1 = 1$, between group $Pennsylvania$ and group $Antidepressants$, $E_2 = 5$, between $Pennsylvania$ and $Antibiotic$, $E_3 = 4$, between $California$ and $Antidepressants$ and $E_4 = 1$, between $California$ and $Antibiotic$ at level L_2. Depending on the selection of split points, the values of $E1, E2, E3, E4$ are varied, but only the maximum value among them, $\max\{E1, E2, E3, E4\}$, decides the maximum influence of these four level L_{i-1} subgraphs which in turn decides the level sensitivity. We use $s = \max\{E1, E2, E3, E4\}$ to represent a split option for a subgraph, namely the selection of a pair of left and right split points. If all the possible split options for a subgraph, which can be generated by randomly or uniformly selecting the pairs of split points, are denoted by $\cup Split_k$, we will have $s \in \cup Split_k$. Second, to minimize the influence of the four subgraphs at level L_{i-1}, the minimum s need to be selected from $\cup Split_k$ in a differentially private manner while splitting the subgraph at level L_i. In order to preserve differential privacy, *DiffPar* employs an exponential mechanism with the utility function designed as:

$$u(subgraph, \cup Split_k) = \frac{1}{s - min(\cup Split_k) + 1}$$

where $min(\cup Split_k)$ denotes the minimum s within $\cup Split_k$. Since $s \geq min(\cup Split_k)$, the maximum change of u happens when s changes from $min(\cup Split_k) + 1$ to $min(\cup Split_k)$, thus giving the sensitivity $\triangle u = 1 - \frac{1}{2} = \frac{1}{2}$. The selected s represents the highest contribution of the 4 subgraphs at level L_{i-1} after splitting, denoted by $senSub$. Finally, the first two steps are repeated for all the subgraphs at level L_i to split each of them to 4 smaller subgraphs at level L_{i-1} while minimizing the influence of all the subgraphs at level L_{i-1}. Once all level L_i subgraphs are split to level L_{i-1} subgraphs, the maximum number of edges contained by a level L_{i-1} subgraph, namely $max(senSub)$, naturally becomes the group-level sensitivity of level L_{i-1} as it would be the maximum influence caused by changing any one subgraph at level L_{i-1}.

Algorithm 1: DiffPar

Input : Bipartite graph BG, privacy budget ϵ and number of specializations n.
Output: Partitioned bipartite graph $\widehat{BG}$ (the $subN$), level sensitivities $senN$.

```
1  Sort both left and right sides based on one attribute;
2  Initialize senN, subN to record sensitivities and subgraphs for specializations;
3  subN(0) ← BG;
4  ε' = ε/n;
5  for i = 1 to n do
6      Initialize senSub to record subgraph sensitivities;
7      for each subgraph ∈ subN(i − 1) do
8          Determine ∪Split_k;
9          Select s ∈ ∪Split_k ∝ exp(ε'u / 2△u);
10         senSub ← s;
11         Split this subgraph with s;
12         subN(i) ← split results;
13     end
14     senN(i) ← max{senSub};
15 end
```

In *DiffPar* (Algorithm 1), initially, after sorting the bipartite graph (line 1), $senN$ and $subN$ are initialized to record sensitivity and subgraphs for each specialization respectively (line 2). Specially, $subN(0)$ records the input BG as the graph without specialization (line 3). After that, the entire privacy budget is equally divided (line 4) for the n specializations (line 5 to 15). Within one specialization, the $senSub$ is first initialized to record the subgraph sensitivities (line 6). Then, for each subgraph in the current level before the nth specialization (line 7 to 13), a set of split options is determined (line 8), where the option s is selected through exponential mechanism (line 9). After recording s in $senSub$ (line 10), we split this subgraph with the pair of split points in option s (line 11) and record the split results in $subN(i)$ (line 12). After we collect $senSub$ from all the subgraphs in this level, the maximum one is the sensitivity of this level (line 14). We next show that the algorithm is differentially private.

Theorem 1: DiffPar is ϵ-differentially private.

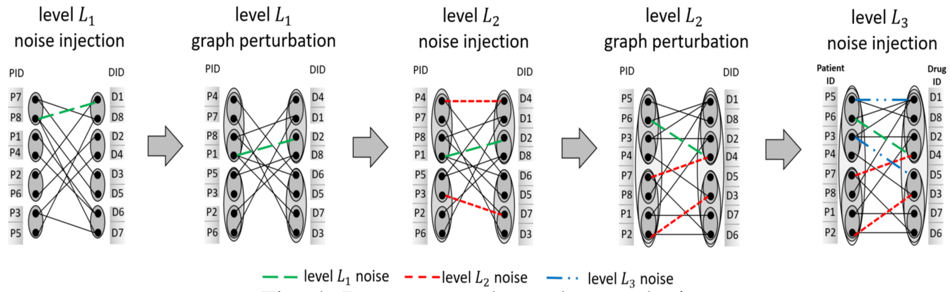


Fig. 4: Bottom-up sub-graph perturbation

Proof: In line 4, the entire budget is divided into n parts for the n specialization based on sequential composition[23]. To achieve ϵ-differential privacy, we need each specialization to guarantee $\frac{\epsilon}{n}$-differential privacy. For each specialization, based on parallel composition [23], the splittings of subgraphs share the same budget. Therefore, each subgraph splitting should preserve $\frac{\epsilon}{n}$-differential privacy, which is guaranteed by the use of Exponential Mechanism[22] in line 9.

B. Bottom-up Sub-graph Perturbation

In the bottom-up sub-graph perturbation process, the partitioned graph produced by *DiffPar* is perturbed through a proposed mechanism called *DiffAggre* that implements a carefully calibrated noise injection and structure-preserving graph perturbation to guarantee group privacy at each hierarchy level. Before presenting the details of noise calibration and structure-preserving subgraph perturbation process in *DiffAggre*, we briefly review its design goals.

1) Design Goal: The goal of the sub-graph perturbation and noise calibration is to protect the inference of the edges between different sub-groups of the exposed differentially private graph under the guarantees of ϵ_g-group differential privacy. Precisely, when a differentially private perturbed output graph is published, a data user accessing the graph at a given access privilege level should not be able to infer any aggregate information in terms of the edges between the subgroups at any granularity finer than what is entitled to the user. For example, if the user's access privilege is at level, L_2 in Figure 3, the data user may be able to access information at the level L_2's subgroup granularity such as the total number of purchases of Antidepressant drugs made by Pennsylvania buyers. Additionally, the data user may also be able to obtain the structural properties (e.g., queries related to the edge distribution within the subgraph) of the subgraphs at level, L_2. However, the user at level, L_2 should not be able to infer any finer level information like the number of purchases of SSRIs antidepressants purchased by buyers in Pittsburgh, which is only entitled to data users of level, L_1.

A key property that we require in the noise injection process of *DiffAggre* is that the noise that is added to guarantee the group differential privacy requirements of the lower levels should be reusable for the higher levels so that the overall noise at the higher levels can be minimized. This motivates the use of a Gaussian Mechanism in *DiffAggre* instead of a Laplace mechanism as it is true for any two Gaussian-distributed random variables $X \sim G(\mu_X, \sigma_X^2)$ and $Y \sim G(\mu_Y, \sigma_Y^2)$, the sum of them $Z = X + Y \sim G(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2)$ also follows Gaussian distribution. This property of Gaussian distribution facilitates the addition of Gaussian noise at each access privilege level during the perturbation process.

2) Graph Structure-preserving Noise Injection: The noise injection process adds a carefully calibrated random Gaussian noise in terms of the number of noisy edges to the grouped graph at various hierarchical levels. Figure 4 shows an example of the noise injection process in *DiffAggre*. The graph in Figure 4 represents the output partitioned graph provided by the top-down partitioning in *DiffPar*(Figure 3). The noise injection in Figure 4 starts from groups in level L_1 and ends at groups in level L_3.

A key challenge in the noise injection process is to ensure that the perturbed noise added graph still retains the structural properties of the original graph. For example, in a drug-purchase association graph, the node distribution within a subgroup of buyers would represent information about the buying trends of the top buyers in that group. If such a group after noise injection loses the structural properties, then a data user trying to obtain the node distribution statistics about the buying trends of the top buyers will not be able to obtain that information. To achieve a structure-preserving noise injection, *DiffAgree* employs the dK-graph model [21], which uses degree correlations of subgraphs to represent the graph structure. dK captures the structure of a graph at different levels of detail into statistics called dK-series [29], [25]. The dK-series is the degree distribution of connected components of some size K within a target graph. For example, $dK - 1$ captures the number of nodes with each degree value, i.e. the node degree distribution. $dK - 2$ captures the number of 2-node subgraphs with different combinations of node degrees, i.e. the joint degree distribution. When $dK - 2$ graph model is used, a graph is described by the edges within it where each edge is represented by the degree of its two terminals. For a subgraph, its DK-series are first extracted to represent it, which contains both the information of number of edges and graph structure (node/edge degrees) that need to be retained during the noise injection. The noise injection process then calibrates a deterministic noise in terms of the number of edges that need to be injected into the subgraphs. Based on Definition 9 on Gaussian mechanism, once δ is decided, the value of c can be calculated, which determines the variance of Gaussian distribution with ϵ and sensitivity $\triangle_2 f$. Therefore, each subgraph samples a random variable following Gaussian distribution $Gaus(c\triangle_2 f/\epsilon)$ and it is calibrated as the number of edges that need to be injected to perturb the number of edges within this subgraph.

We present the pseudo-code of the Botton-up Aggregation, *DiffAggre* algorithm in Algorithm 2. Initially, V is initialized to record the variances for all the levels (line 1). Then, for each level (line 2 to 18), the sensitivity is selected (line 3) to calculate the variance (line 4 to 5). From line 6 to 16, the noises are injected for several times. In each time, the aggregated noises can be reused to reduce the variance (line 8 to 10) first. After that, the reduced variance is used to inject noise to each subgraph through Gaussian Mechanism (line 11 to 14). Once all the noises have been generated, the new variances are recorded in V (line 15). After noise injection, the graph perturbation is implemented (line 17).

Algorithm 2: DiffAggre

Input : Partitioned bipartite graph $\widehat{BG}$, privacy budget group and structure ϵ_g, ϵ_s, the sensitivities for each specialization $senN$, the total number of levels n, the required number of specializations for each level Spe.
Output: Perturbed bipartite graph $\widetilde{BG}$.

1 Initialize V to record the variances for all the levels;
2 **for** $i = 0$ to $n-1$ **do**
3 $\quad sen = senN(Spe(i))$;
4 $\quad \delta = \frac{sen}{\epsilon_g}\frac{n(n-1)}{2}$;
5 $\quad \delta^2_{real} = \delta^2$;
6 $\quad$ **for** $j = n$ to $i+1$ **do**
7 $\quad\quad$ Initialize a list Var to record the variances;
8 $\quad\quad$ **for** $t = 1$ to j **do**
9 $\quad\quad\quad \delta^2_{real} = \delta^2_{real} - aggre\{V_t\}$;
10 $\quad\quad$ **end**
11 $\quad\quad$ **for** $each\ subgraph\ in\ level\ j$ **do**
12 $\quad\quad\quad noise = Gau(\delta_{real})$;
13 $\quad\quad\quad$ record δ^2_{real} in Var;
14 $\quad\quad$ **end**
15 $\quad\quad$ record Var in V_j
16 $\quad$ **end**
17 $\quad Pert(level\ (i+1))$;
18 **end**

Theorem 2: DiffAggre is (ϵ, δ)-differentially private.

Proof: In *DiffAggre*, there are $\frac{n(n-1)}{2}$ possible group-differentially private bipartite graph disclosures. Based on parallel composition, all the subgraphs in the same level share the same budget. So the entire budget ϵ can be divided into $\frac{2}{n(n-1)}\epsilon_g$ fractions to be used by the Gaussian Mechanism, making the entire *DiffAggre* process differentially private .

IV. Experimental Evaluation

In this section, we experimentally evaluate the performance of the proposed group differential privacy-aware data disclosure algorithms. Before presenting the results, we first briefly describe the experimental setup.

Datasets	Left-side node	Right-side node	Edges
Amazon	1851132	252331	2982326
Song	992	1084620	4413834
Movie	69878	10677	10000054

TABLE IV: Summary of bipartite graph datasets

A. Experimental setup

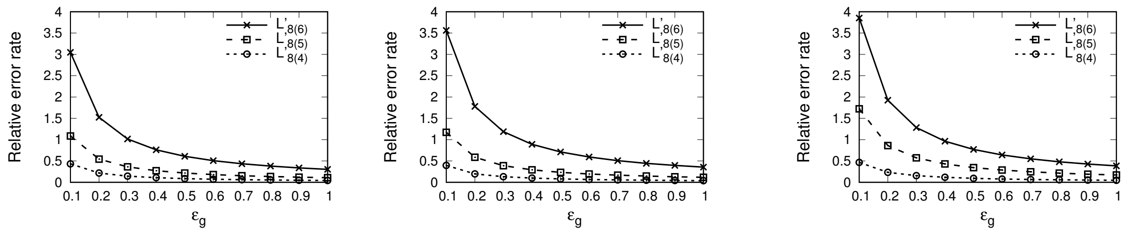
The proposed differentially private partitioning and graph perturbation algorithms were implemented in Java with an Intel Core i7 2.70GHz PC with 16GB RAM and evaluated using three datasets including the Amazon product review dataset [16], Last.fm songs dataset [5] and MovieLens 100k dataset [15] (Table IV). The Amazon dataset consists of nodes representing users and products in health and personal care category and edges represent the individual ratings. The Song dataset has users as the left-side nodes and songs as the right side nodes and edges represent individual ratings. The Movie dataset describes ratings of movies (right-side nodes) made by users (left-side nodes).

B. Experimental results

Our experimental evaluation consists of three parts. First, we evaluate the amount of noise required for protecting ϵ_g-group differential privacy for different protection levels using the three datasets. Then, the performance of *Diffpar* and *DiffAggre* to protect various ϵ_g-group differential privacy levels is analyzed. Finally, we study the impact of varying the specialization depth on the obtained results. We use relative error rate (RER) as a metric to measure the accuracy of the disclosed differentially private data. For level L_i with m subgraphs, we can first calculate absolute error $AE_j = |PC_j - TC_j|$ ($j \in [1, m]$) for each subgraph, where PC_j denotes perturbed edge count and TC_j denotes true edge count. Then, relative error rate is calculated as $RER = \frac{\sum_{j=1}^{m} AE_j}{Total}$, where $Total$ represents true edge count of the entire bipartite graph. Without any privacy protection, with no noise injected, $RER = 0$. Thus, a lower RER represents higher data utility as more accurate information can be retained in the published data.

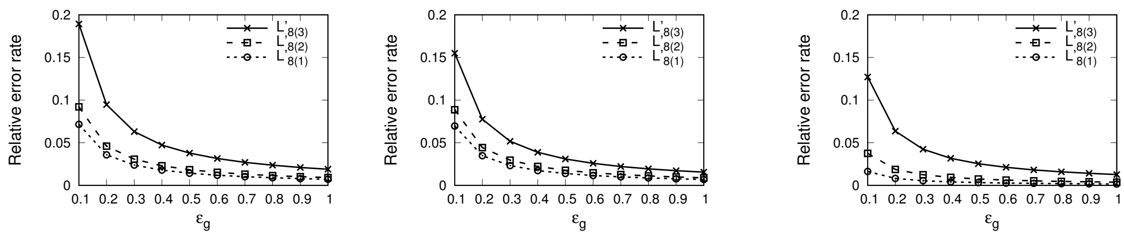
1) Impact of ϵ_g with varying group protection levels: Our first set of experiments evaluates the amount of noise required for protecting various group protection levels. Specifically, for each bipartite graph, we run *DiffPar* to do seven specializations so that eight levels, denoted by $L_i (1 \leq i \leq 8)$ are formed, where L_8 represents the original graph without partitioning and L_1 contains the most fine-grained subgraphs. Here, adjacent levels differ by only one specialization (similar to the example shown in Figure 3). By injecting different amount of noise into the original L_8 bipartite graph, different levels, from L_6 to L_1, can be protected. Intuitively, with less noise (lower RER), ϵ_g-group differential privacy can be achieved for lower levels with finer-grained subgraphs. However, with higher noise (higher RER), higher levels with coarser-grained groups can also be protected. We measure the relative error, RER for six possible disclosures $L'_{8(j)} (1 \leq j \leq 6)$ that represent fixed disclosure level L_8 and varying group protection levels $L_j, 1 \leq j \leq 6$. Here, the privacy budget for *DiffPar* is set to 1 while the privacy budget for one level noise injection in *DiffAggre* is varied from 0.999 to 0.1. The value of δ is set to 0.001 for all the experiments.

The results for achieving ϵ_g-group differential privacy for three higher levels L_6, L_5, L_4 with coarse-grained subgraphs is shown in Figure 5 and the results for the three lower levels L_3, L_2, L_1 with fine-grained subgraphs is shown in Figure 6. Here, all the three datasets (A=Amazon, S=Song, M=Movie) are used and compared. First we observe that for all datasets, RER for $L'_{8(a)}$ is always higher than RER for $L'_{8(b)} (b \leq a)$ which shows that more noise is required to protect group differential privacy for higher levels. The reason is that sensitivity for higher levels is always higher than sensitivity for lower levels. Second, we note that when ϵ_g is varied, smaller ϵ_g makes RER larger. Specifically, when $\epsilon_g = 0.999$, all the disclosures from $L'_{8(1)}$ to $L'_{8(6)}$ show small relative error and $L'_{8(1)}$ generates RER less than 1% for all tested datasets. Their RER upper-bound increase to 17% at $L'_{8(5)}$ and finally reaches 35% at $L'_{8(6)}$. As can be



(a) Amazon (b) Song (c) Moviw

Fig. 5: Impact of ϵ_g for protecting coarse-grained subgraphs



(a) Amazon (b) Song (c) Movie

Fig. 6: Impact of ϵ_g for protecting fine-grained subgraphs

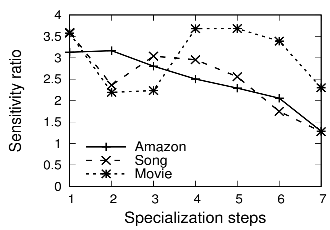
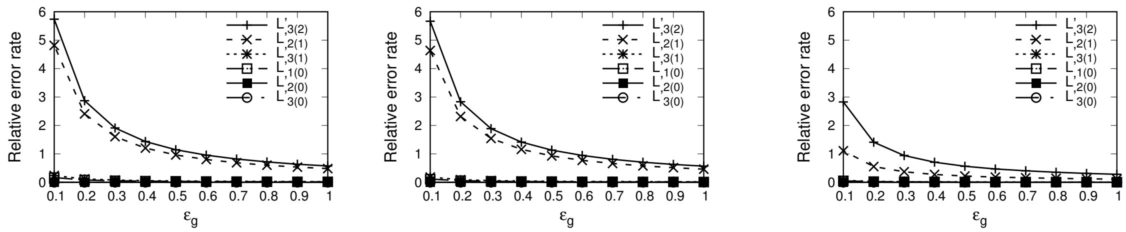


Fig. 7: Sensitivity ratio after specializations

seen, users accessing level L_8 with noise protecting level L_6 are given highly perturbed information. After that, as ϵ_g is decreased, RER for all the disclosures gradually increases. When ϵ_g goes down to 0.1, the budget is highly restricted and hence more noise has to be injected. It makes the RER for all the disclosures increase significantly, especially for $L'_{8(6)}$ and $L'_{8(5)}$. However, their RER upper-bound reduces for $L'_{8(4)}$ and $L'_{8(3)}$ to be 5% and 2% respectively to provide higher utility for users with higher privilege. Finally, by comparing the results from the three different datasets, we found that the Amazon dataset and the Movie dataset show highest and lowest RER respectively for $L'_{8(6)}$, $L'_{8(5)}$ and $L'_{8(4)}$ in Figure 5. However, it is interesting to find that their performance is opposite for $L'_{8(3)}$, $L'_{8(2)}$ and $L'_{8(1)}$ in Figure 6. The difference is mainly impacted by sensitivity, which is related to the features of the datasets. As we have discussed, each specialization splits a subgraph into four smaller sub graphs and therefore reduces the sensitivity since the maximum contribution of the smaller subgraphs is usually smaller than the original parent subgraph. We measure sensitive ratio that captures the reduction of the sensitivity after the specialization in comparison to the sensitivity before specialization. Ideally, the sensitivity ratio after each specialization is 4. However, due to the error introduced in the exponential mechanism in partitioning, sensitivity ratio is always smaller than 4 in practice. We present the sensitivity ratio of all seven specializations for the three datasets in Figure 7. As can be seen, during the first three specializations, Amazon dataset has the highest ratio while Movie dataset has the least ratio. The key reason is due to the difference between the average node degree of the datasets. With much higher average node degree, Movie dataset is more influenced by the skewed node degree distribution, which results in smaller sensitivity ratio that does not get adjusted fast during the first few specializations. However, during the last four specializations, because of highly partitioned groups and larger volume of edges, sensitivity ratio of Movie dataset becomes high, which results in smaller sensitivity and lower RER in Figure 6(c).

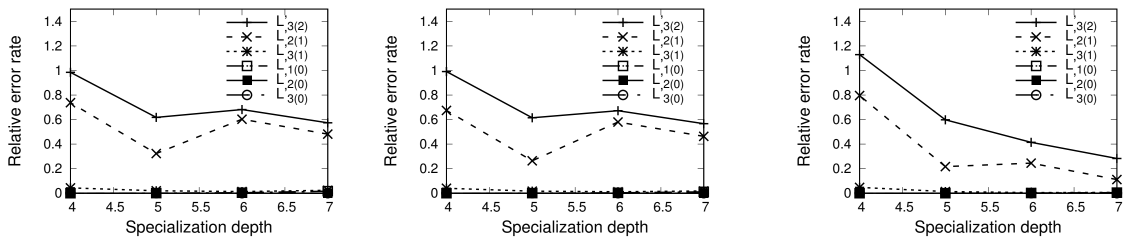
2) Impact of ϵ_g for varying Group Disclosure levels: This set of experiments evaluates the performance of the *DiffPar* and *DiffAgree* algorithms to protect group-differential privacy at various group disclosure levels with varying ϵ_g. For this experiment, the *DiffPar* partitions the entire bipartite graph into 3 levels through 7 specializations, where level L_3 is the entire bipartite graph while level L_2 and L_1 are generated through 3 and 7 specializations respectively. In addition, we consider traditional differential privacy protection for the inference of a single edge at the lowest level L_0 with group size 1. Therefore, there are six possible disclosures, namely $L'_{3(0)}$, $L'_{2(0)}$, $L'_{1(0)}$, $L'_{3(1)}$, $L'_{2(1)}$ and $L'_{3(2)}$ and the privacy budget for *DiffPar* is set as 1 while ϵ_g for noise injection in DiffAggre is changed from 0.999 to 0.1. The RER values of the six disclosures with varying ϵ_g are measured.

The results for three datasets are shown in Figure 8. First, it is clear that there is a huge gap of RER between $L'_{2(1)}$, $L'_{3(2)}$ and other disclosure levels. A larger RER for $I_{a,b}$ can be attributed to two factors, namely sensitivity of level L_b and number of subgraphs at level L_a. For $L'_{3(0)}$, $L'_{2(0)}$ and $L'_{1(0)}$, level L_0 has sensitivity as low as 1. For $L'_{3(1)}$, although sensitivity for level L_1 becomes larger, there is only 1 graph at level L_3. Therefore, RER increases significantly. From this perspective, $L'_{2(1)}$ has both higher L_1 sensitivity and a larger number of subgraphs in L_2 while $L'_{3(2)}$ has high L_2 sensitivity,



(a) Amazon (b) Song (c) Movie

Fig. 8: Impact of ϵ_g for varying Group Disclosure levels



(a) Amazon (b) Song (c) Movie

Fig. 9: Impact of depth of specialization

which results in larger RER. Second, we can see that the Movie dataset shows significantly lower RER for all values of ϵ_g for all disclosures compared to Amazon and Song datasets. For $L'_{3(0)}$, $L'_{2(0)}$ and $L'_{1(0)}$, theoretically, all datasets suffer from the same amount of noise as L_0 sensitivity is always 1. However, since Movie dataset has the largest number of edges, RER of Movie dataset becomes lower. For $L'_{3(1)}$ and $L'_{2(1)}$, as we have discussed, Movie dataset has the smallest L_1 sensitivity after 7 specializations. It results in lower RER at these two disclosures. For $L'_{3(2)}$, from Figure 7 we can find that Movie dataset has higher sensitivity after 3 specialization and therefore higher L_2 sensitivity. As a result, Movie dataset has the largest difference between RER of $L'_{3(2)}$ and RER of $L'_{2(1)}$ because of its higher L_1 sensitivity among the three datasets.

3) Impact of depth of specialization: This set of experiments evaluates the performance of *DiffPar* and *DiffAgree* algorithms by varying the depth of specialization of the disclosed association bipartite graph. For this experiment, we consider levels L_3 to L_0 but we vary the specialization depth, namely the number of specialization steps required by *DiffPar* to generate subgraphs at L_1. We denote the specialization depth as d. Then, L_1 requires d specializations and L_2 requires $\lfloor d \rfloor$ specializations. The privacy budgets for *DiffPar* is set to 1 and ϵ_g for *DiffAggre* is set to 0.999. The relative error rates, RERs for $L'_{3(0)}$, $L'_{2(0)}$, $L'_{1(0)}$, $L'_{3(1)}$, $L'_{2(1)}$ and $L'_{3(2)}$ are measured.

The results for the three datasets are shown in Figure 9. As can be seen, Amazon and Song datasets have quite similar performance, which is very different from that of the Movie dataset. Movie dataset has the highest RER at $d = 4$. This can be explained by the change of sensitivity ratio in Figure 7. The sensitivity ratio of Movie dataset is the lowest after 3 specializations and then becomes higher during the later 4 specializations. Therefore, the last four specializations primarily improve the performance for Movie dataset. When specialization depth is low, none or few last four specializations can be involved in, which results in lower performance. However, by increasing specialization depth from 4 to 7, all the last four specializations can be included and hence the Movie dataset demonstrates the best performance.

V. RELATED WORK

The problem of information disclosure has been studied extensively in the framework of statistical databases. Samarati and Sweeney [26],[27] introduced the k-anonymity approach which has led to some new techniques and definitions such as l-diversity [20] and t-closeness [19]. There had been some work on anonymizing graph datasets with the goal of publishing statistical information without revealing information of individual records. Backstrom et al. [2] show that in fully censored graphs where identifiers are removed, a large enough known subgraph can be located in the overall graph with high probability. Ghinita et al. present an anonymization scheme for anonymizing sparse high-dimensional data using permutation based methods [14] by considering that sensitive attributes are rare and at most one sensitive attribute is present in each group. The safe grouping techniques proposed in [4], [8] consider the scenario of retaining graph structure but aim at protecting privacy when labeled graphs are released. But, as mentioned earlier, these existing schemes have been focused on individual privacy and do not provide support for group privacy.

Based on the concept of differential privacy[10], there had been many work focused on publishing sensitive datasets through differential privacy constraints [7], [13], [28]. Differential privacy had also been applied to protecting sensitive information in graph datasets such that the released information does not reveal the presence of a sensitive element [9], [17], [25]. Recent work had focused on publishing graph datasets through differential privacy constraints so that the published graph maintains as much structural properties as possible while providing the required privacy [25]. But, as mentioned earlier, these existing schemes do not support group privacy and multi-level access to the published dataset. The preliminary discussion of group privacy proposed in this work is briefly introduced in a recent poster publication by the authors [24]. In this paper, we propose the *Diffpar* and *DiffAggre* algorithms that apply the proposed notion of group differential privacy over bipartite association graph data to provide guaranteed group differential privacy. To the best of our knowledge, our work presented in this paper is the first significant effort on providing guaranteed group privacy and multi-level group privacy protection in a large-scale dataset such as association graphs.

VI. Conclusion

Existing privacy-preserving data publishing techniques have primarily focused on protecting the privacy of individual's information with the assumption that all aggregate (statistical) information about individuals are safe for disclosure. In this paper, we have focused on scenarios when aggregate information about a group of individuals can be sensitive and needs protection. We proposed the notion of ϵ_g-*Group Differential Privacy* and studied the problem of group privacy protection in the context of bipartite association graphs. We developed a suite of differentially private mechanisms that guarantee group privacy requirements of users, allowing data users to obtain different levels of information based on the group privacy protection levels in the disclosed data. Extensive experiments on real association graph data show that the proposed techniques are effective, efficient and provide the required level of privacy.

Reference

[1] Bigdata and future of privacy. https://epic.org/privacy/big-data/.

[2] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th international conference on World Wide Web*, pages 181–190. ACM, 2007.

[3] Michael Batty. Big data, smart cities and city planning. *Dialogues in Human Geography*, 3(3):274–279, 2013.

[4] Smriti Bhagat, Graham Cormode, Balachander Krishnamurthy, and Divesh Srivastava. Class-based graph anonymization for social network data. *Proceedings of the VLDB Endowment*, 2(1):766–777, 2009.

[5] O. Celma. *Music Recommendation and Discovery in the Long Tail*. Springer, 2010.

[6] Hsinchun Chen, Roger HL Chiang, and Veda C Storey. Business intelligence and analytics: From big data to big impact. *MIS quarterly*, 36(4), 2012.

[7] Rui Chen, Noman Mohammed, Benjamin CM Fung, Bipin C Desai, and Li Xiong. Publishing set-valued data via differential privacy. *Proceedings of the VLDB Endowment*, 4(11):1087–1098, 2011.

[8] Graham Cormode, Divesh Srivastava, Ting Yu, and Qing Zhang. Anonymizing bipartite graph data using safe groupings. *Proceedings of the VLDB Endowment*, 1(1):833–844, 2008.

[9] Wei-Yen Day, Ninghui Li, and Min Lyu. Publishing graph degree distribution with node differential privacy. In *Proceedings of the 2016 International Conference on Management of Data*, pages 123–138. ACM, 2016.

[10] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, volume 3876, pages 265–284. Springer, 2006.

[11] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[12] Wei Fan and Albert Bifet. Mining big data: current status, and forecast to the future. *ACM sIGKDD Explorations Newsletter*, 14(2):1–5, 2013.

[13] Arik Friedman and Assaf Schuster. Data mining with differential privacy. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 493–502. ACM, 2010.

[14] Gabriel Ghinita, Yufei Tao, and Panos Kalnis. On the anonymization of sparse high-dimensional data. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 715–724. Ieee, 2008.

[15] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, 2016.

[16] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, pages 507–517. International World Wide Web Conferences Steering Committee, 2016.

[17] Vishesh Karwa, Sofya Raskhodnikova, Adam Smith, and Grigory Yaroslavtsev. Private analysis of graph structure. *Proceedings of the VLDB Endowment*, 4(11):1146–1157, 2011.

[18] Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Analyzing graphs with node differential privacy. In *Theory of Cryptography*, pages 457–476. Springer, 2013.

[19] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 106–115. IEEE, 2007.

[20] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 24–24. IEEE, 2006.

[21] Priya Mahadevan, Dmitri Krioukov, Kevin Fall, and Amin Vahdat. Systematic topology analysis and generation using degree correlations. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 135–146. ACM, 2006.

[22] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 94–103. IEEE, 2007.

[23] Frank 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*, pages 19–30. ACM, 2009.

[24] Balaji Palanisamy, Chao Li, and Prashant Krishnamurthy. Group differential privacy-preserving disclosure of multi-level association graphs. In *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*, pages 2587–2588. IEEE, 2017.

[25] Alessandra Sala, Xiaohan Zhao, Christo Wilson, Haitao Zheng, and Ben Y Zhao. Sharing graphs using differentially private graph models. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 81–98. ACM, 2011.

[26] Pierangela Samarati. Protecting respondents identities in microdata release. *IEEE transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[27] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[28] Qian Wang, Yan Zhang, Xiao Lu, Zhibo Wang, Zhan Qin, and Kui Ren. Rescuedp: Real-time spatio-temporal crowd-sourced data publishing with differential privacy. In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, pages 1–9. IEEE, 2016.

[29] Yue Wang and Xintao Wu. Preserving differential privacy in degree-correlation based graph generation. *Transactions on data privacy*, 6(2):127, 2013.

[30] Xindong Wu, Xingquan Zhu, Gong-Qing Wu, and Wei Ding. Data mining with big data. *IEEE transactions on knowledge and data engineering*, 26(1):97–107, 2014.

[31] Yin Yang, Zhenjie Zhang, Gerome Miklau, Marianne Winslett, and Xiaokui Xiao. Differential privacy in data publication and analysis. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 601–606. ACM, 2012.