Graph-Based Privacy-Preserving Data Publication

Xiang-Yang Li*‡§, Chunhong Zhang†, Taeho Jung§, Jianwei Qian§, Linlin Chen§
* School of Computer Science and Technology, University of Science and Technology of China
† State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
‡ School of Software, Tsinghua University
§ Department of Computer Science, Illinois Institute of Technology, Chicago

***Abstract*—We propose a graph-based *framework* for privacy preserving data publication, which is a systematic abstraction of existing anonymity approaches and privacy criteria. Graph is explored for dataset representation, background knowledge specification, anonymity operation design, as well as attack inferring analysis. The framework is designed to accommodate various datasets including social networks, relational tables, temporal and spatial sequences, and even possible unknown data models. The privacy and utility measurements of the anonymity datasets are also quantified in terms of graph features. Our experiments show that the framework is capable of facilitating privacy protection by different anonymity approaches for various datasets with desirable performance.**

***Keywords*—privacy preservation, data publication, graph partition**

I. Introduction

The problem *privacy-preserving data publishing*(PPDP) [1] has received wide research interests in the past decades. For a user u with identification id and sensitive information s, the privacy is considered leaked when an attacker can infer the pair (id, s) with high certainty according to the published anonymity data. Early techniques of PPDP mainly focus on relational tables stored in database, among them including the well known *k-anonymity* [2], *l-diversity* [3], and data perturbation [4]. The dramatically increasing ability of collecting and analyzing large amounts of data leads to the research of PPDP evolving from relational tables to various data models, such as social networks [5] and temporal/spatial data of Location-Based Service [6], where the data present large diversity in terms of structures and secret types. While the basic notion of privacy, i.e., masking a person in a group where she is indistinguishable from others, is extended naturally to various scenarios, a large body of prior literature has been parallelly explored for particular data model.

A question naturally arises: Is it possible to leverage the existing privacy preservation methods through a general model? Can we uniformly define and assess privacy by this model so as to make comparison among different anonymity approaches become possible and even intuitive? This motivates us to propose a novel framework to facilitate a universal anonymity strategy for various data models. The generality of framework will face several challenges. First, the framework should be capable of representing a variety of data types. In general, datasets may significantly differ in schema, attribute distributions, as well as privacy concerned. From the privacy point of view, the information contained in datasets is generally categorized as follows. i) Identifications of objects(**IoB**): A dataset is commonly organized as a list of multiple distinct objects (e.g, persons or companies). ii) Attributes of objects(**AoB**): The non-sensitive attributes describe the information of the objects and usually make them distinguishable from each other (e.g.,*Age* or *Sex* of a person). The combination of partial or entire of such attributes is typically called *quasi-identifiers* QI [7]. iii) Secrets of Objects(**SoB**): Any sensitive information that the object tends to keep from disclosure,and the leakage of which will damage the object's privacy, is defined as *secret*. Preventing disclosure of SoB is the primary goal of privacy preservation. iv) Relationship between objects(**RoB**): Many datasets naturally possess network structures in which various kinds of relationships between objects are characterized (e.g., friend relationship in social networks, or co-author relationship in paper corpus). RoB can be viewed as special case of AoB or SoB respectively [8].

Second, as privacy threats are closely related to the assumption of adversary's prior belief, the framework should be able to clearly quantify the background knowledge as attacker's capability. There are mainly two kinds of assumption for background knowledge. One ignores the prior belief where every secret is viewed equally possible for every individual (such as *k-anonymity*), the other models the background knowledge with various types and amounts via different specifications [9], [10]. Analogously, flexible privacy protection requirements (e.g.,the disease *flu* is not thought the same sensitive as *HIV*) also need to be taken into account. These properties allow the framework to prevent potential *over-protection* by restricting the assumption of attacker capability, as well as unreasonable ignorance of threats in practice.

Third, the *utility metric* of the framework should be entirely independent of specific anonymity approaches and be generally quantified from the perspective of data usage. It is common in previous works that the utility is measured by metrics highly related to privacy algorithms and parameters. For example, the size k of equivalent class in *k-anonymity* is used by utility definition of discernibility [11]. While the metrics of statistical information, such as *mean* and *correlation*, of anonymous datasets are concerned by perturbation approaches [4], it is not clear whether we can extend them to other anonymity approaches. Therefore, a *privacy independent* utility metric is necessary to provide paradigm for comparing different privacy strategies.

In this work, we propose a *graph-based* privacy preservation

framework for data publication. The graph is introduced as basic methodology for data representation (§II-A), adversary capability modeling (§II-B), privacy and utility measurement metric (§III-D), and graph partition algorithm for different privacy approaches(§IV). The idea to relate dataset to graph is not entirely new, nevertheless to our best knowledge, there have been rarely research efforts in the direction of graph-based privacy preservation framework. Although the relational database can often be transformed into an information network, our framework is much more general and can handle various datasets.

A distinguished advantage of our framework is that we can explicitly encode inherent relevance relationship of data objects into graph structure, on which the two antagonistic sides, attack and protection, of privacy preservation can be coherently articulated. The privacy breach risk as well as utility evaluation can also be efficiently characterized by graph properties, which permits placing different anonymity approaches on a comparable base. The contributions of this paper include:

- We propose a novel graph representation feasible for original dataset, anonymity dataset, as well as background knowledge of adversary;
- We define graph-based privacy criterion and utility metrics, which are used to compare the efficiency of different anonymity approaches on relational tables, social networks, and temporal and spatial datasets by experiment.
- We propose a spectrum graph partition algorithm as a universal method to construct equivalent class as subgraph within which the existing anonymity operations are independently performed.

The rest of this paper is organized as follows. Section II illustrates general graph construction principle and section III thoroughly discuss the main building blocks of the framework. Section IV describes the algorithms of graph partition. Experiments on three datasets are shown respectively in section V. We review the related work in Section VI and conclude the paper in Section VII.

II. Graph Model

A. Data Graph

The first problem of framework we face is how to construct a general graph for various datasets. Given a dataset T with arbitrary structures (*e.g.*, a tabular dataset, a social network structure), we use a *data graph* $G = \{U \cup V \cup S, E\}$ to model the objects by *vertices* and relations by *edges*. In particular, each individual user is abstracted as a single *user vertex* $u \in U$, whereas each *unique quasi-identifier* attribute value appeared in T is abstracted to an *attribute vertex* $v \in V$, and each individual secret corresponds to a *secret vertex* $s \in S$. We represent an attribute node as *type:value*, where $type$ and $value$ are the attribute's type and value. The union of V and S is also called **alphabet** $\Sigma = V \cup S$ of graph G whose cardinality $|\Sigma|$ is the number of unique attribute values contained in dataset T. Note that the vertices of graph are now just applied to categorical and discrete numerical data where the attributes have finite domains. The data of continuous-valued attributes which could be approximated by optimal discretization with bounded information loss are beyond the scope of this paper.

The edge of graph G indicates two types of relations between vertices: *semantic* relation and *correlation* relation. An edge e_{ij} is called *semantic edge* if its two endpoints have *same* vertex type. For instance, an edge connecting user vertices *Alice* and *Bob* indicates their friendship in social network, an edge between attribute vertices *age:20* and *age:30* reflects the distance in the domain of attribute *age*. Naturally, the edge weight ω_{ij} of edge e_{ij} is selected as the *semantic similarity* between two vertices. On the other hand, an edge connecting vertices with *different* types represents their *correlation* and therefore termed as *correlation edge*. For example, a user vertex u will connect all its attribute vertices $\{v_i\}$ and secret vertices $\{s_j\}$ by correlation edges of weight 1. Fig.1 shows an example of data graph G. A dataset T can be entirely represented by graph G defined as following without information loss.

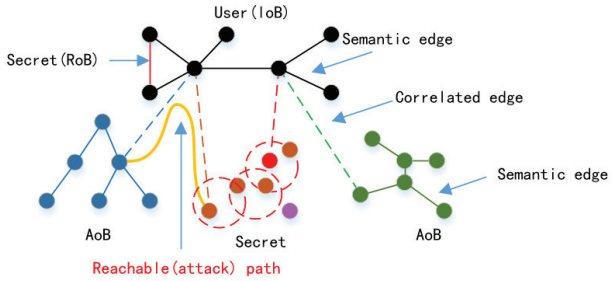

Fig. 1. Data Graph G: solid edges are semantic edges connecting nodes of same type vertices; dashed edges are correlation edges connecting nodes of different type vertices. Black nodes are IoB; green and blue nodes are AoB; other nodes are SoB; one edge between users is secret ROB.

Definition 1: A **Data Graph** is a graph $G = (U \cup \Sigma, E)$ for dataset $T = \{ID, A, R\}$ composed by sets of *identification ID*, *attribute* A and *relation* R. The mapping function $\Phi(T) = \{\phi_U, \phi_A, \phi_R\}$ converts objects of dataset T to elements of graph G:

- Function $\phi_U : ID \mapsto U$ maps user identification id to user vertex u, i.e., $u = \phi_U(id)$;
- Function $\phi_A : A \mapsto \Sigma$ maps attribute a to attribute vertex v or secret vertex s, *i.e.*, $v/s = \phi_A(a)$;
- Function $\phi_R : R \mapsto E$ maps relation r to weight of edge e, *i.e.*, $\omega_e = \phi_R(r)$.

Similarly, the reverse function $\Phi^{-1}(G) = \{\phi_U^{-1}, \phi_A^{-1}, \phi_R^{-1}\}$ can convert graph G to dataset T without information ambiguity. Consequently, the properties of G and T are same, where an operation on vertices or edges of graph G can be associated to certain operations on elements of dataset T. Thus the privacy guarantees of G and T are guaranteed to be equivalent to each other.

B. Attack Graph

The data graph G captures the intrinsic properties of dataset T. Likewise, the background knowledge, denoted as BK, of adversary can be also modeled as **Attack Graph** $G_A = \{ID_A \cup V_A \cup S_A, E_A)$, where ID_A, V_A and S_A are vertex sets

of *identification*, *attribute* and *secret* respectively. The vertex sets of G_A are allowed to be different to those of data graph G since adversary may obtain BK from various sources. In general, the more overlap between G_A and G, the more attack capability the adversary has.

The main difference of G_A with G arises from the construction principles of edges E_A. As the adversary is not interested in data utility, semantic edges between attribute vertices are in general not necessary and the correlation edge is the only type in E_A. One exception is for semantic edge between nodes of ID_A, which are used to indicate the non-sensitive or secret relationships between users. If all the semantic edges are absent, G_A is simplified to a bipartite graph where the edge can only exist between nodes of ID_A and $\Sigma_A = V_A \cup S_A$.

In designing the attack graph G_A, we must provide means to explicitly specify the background knowledge BK. Although it is generally impossible for data publisher to know the precise BK of arbitrary adversary, the type and amount of BK can be instead assumed reasonably. Let ω_{ij} (omit the notation of *adversary* for brevity) be the edge weight of e_{ij} connecting user node id_i and secret node s_j in attack graph G_A. The basic unit BK that adversary thinks user id_i has secret s_j with probability p_{ij} is explicitly expressed by assigning $\omega_{ij} = p_{ij}$, which following the basic idea of [9]. Special cases are the BKs of positive associations and negative associations [10], that is, id_i has secret s_j, or id_i does not have secret s_j, the corresponding ω_{ij} is then set to 1 or 0 respectively. More complex expression of BKs can be usually represented by consistent-setting of weights of related edges. For correlational knowledge [9] such as "the prevalence of *cancer* was higher for users of Age 50 than those of Age 20", the order constraint $\omega_{(id_1 \text{with age 50,caner})} > \omega_{(id_2 \text{with age 20,cancer})}$ will be hold to satisfy this belief. This principle of consistent-setting can be naturally applied to BKs with similar form such as (id_1 *has* s_i) $\rightarrow$ (id_2 *has* s_j) [12]. For the knowledge of same-value families, i.e., if user id_1 of family $F = \{id_1, \ldots, id_k\}$ has *college education*, than the other users of F also tend to have the same education degree. This belief can be represented by setting all the $\omega_{(id_k, college)}$ of F equal to each other.

The selection of BK reflects the assumption for attacker's capability. Some existing knowledge representations, such as knowledge graph, naturally provide sources of various assumptions for G_A construction [13]. In addition, the original graph G is always viewed as a good source to extract BK of both data distribution and particular individuals [9], [14]. For instance, a simple G_A can be directly obtained by extracting a subgraph from original data graph G, i.e., $G_A \subseteq G$.

III. Privacy-Preserving Framework

A. Architectures

We now sketch the graph based privacy preservation framework $\mathcal{F}$ shown in Fig.2. Our framework $\mathcal{F}$ is modeled as a collection of function modules along with interfaces among them. For a dataset T to be published with privacy requirements, the framework $\mathcal{F}$ first converts T to data graph G by a mapping function $G = \Phi(T)$. The graph G is then partitioned into a set of subgraphs $\{g_i \mid i = 1, 2, \cdots\}$ by a procedure $\Pi(G)$ (presented in next section) to construct anonymity *equivalence classes*. The optimization constraints of graph partition arise from the tradeoff between *privacy* guarantee and high data *utility*. Subsequently, the subgraphs $\{g_i \mid i = 1, 2, \cdots\}$ are taken as input of anonymization building block $\mathcal{A}$ to acquire **Anonymity Graph** G^*, which is obtained by using a group of *anonymity operators* to graph G (different subgraphs may need different privacy anonymization methods). The privacy requirements are posed by data publisher as a set of privacy principles or explicit privacy guarantees for each user. In parallel, the attack graph G_A is generated by data publisher according to her assumption of background knowledge. The risk of privacy leakage of G^* is evaluated upon **disclosure graph** H, which is inferred by seeing G^* and G_A together. The utility of G^* is also computed according to graph-based utility metrics U_E and U_C respectively. As long as the evaluations violate the privacy requirements, the process iteratively goes back to the graph partition. Adjustments on attack graph G_A choice or tradeoff setting between privacy guarantee and utility impact the evaluations success. Finally, the eligible anonymity graph G^* is converted to dataset T^* by inverse function Φ^{-1}, then published.

B. Graph Anonymity Operators

The basic idea of most privacy protection approaches in literature is to modify the dataset such that a particular user is indistinguishable with some others, or the association between user and her secret is broken. We propose *anonymity operator*, a sequence of *vertex/edge addition/deletion*, to generalize various privacy protection methods. That is, a graph G can be modified to *anonymity graph* G^* by a sequence of anonymity operators, referred as $G \rightarrow G^*$.

For graph G, *vertex perturbation* is defined as the modification to its alphabet Σ. Any elements of Σ can be deleted or new ones can be added. The operators of vertex addition and deletion are denoted as v^+ and v^- respectively. Any modification about vertices can then be decomposed into an *operator sequence* (PS) of v^+ and v^-. The primary vertex perturbations include: *Generalization* (*e.g.*, change age:20 to age:[10-30]), *Randomization* (*e.g.*, change a vertex to set of vertices and assign a probability distribution to these vertices), *Aggregation* (grouping a set of vertices to a new virtual vertex), and *Permutation* (shuffling a subset of vertices),etc..

Similarly, the *edge perturbation* operators, denoted as edge addition e^+ and deletion e^-, converts edge set E to anonymized E^*. Any modification on edge can also be decomposed into an operator sequence of e^+ and e^-. The typical edge perturbations include: *Changing* of endpoints of edge, *Modification* of edge weight, *Aggregation* of a group edges, and *Division* of an edge e into a group edges (which needs proper weight assignment on edges). A particular perturbation approach can be expressed by the combination of vertex perturbation and edge perturbation. An instance of anonymity operators for three anonymity approaches are shown in Fig.3.

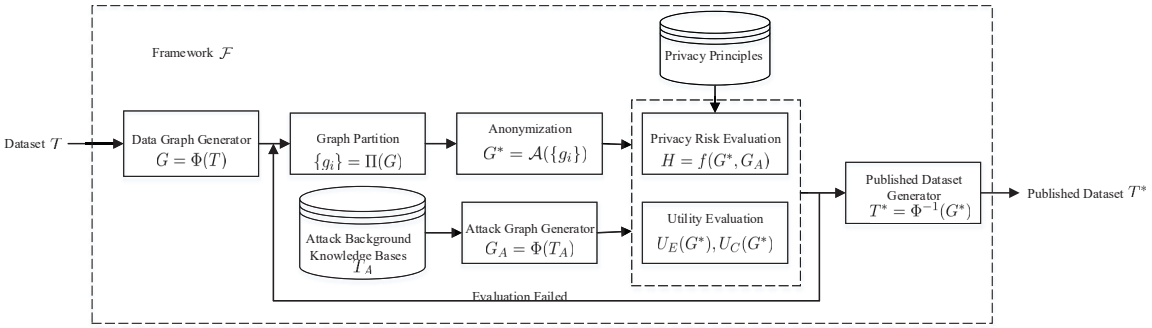

Fig. 2. Graph-Based Privacy-Preserving Data Publishing Framework $\mathcal{F}$

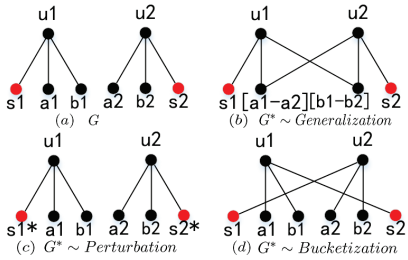

Fig. 3. Anonymity operators sequence for three anonymity approaches. (a) In original data graph G, users u_1 and u_2 have two attributes and one secret. (b) The anonymity graph G^* is produced by *generalization* [1] approach such as *k-anonymity*. The attribute vertices a_1 and a_2 are generalized to a new vertex labeled by range $[a_1 - a_2]$. This outcome can be viewed as the results of vertex perturbation operator sequence $PS = (a_1^-, a_2^-, [a_1 - a_2]^+)$ and edge perturbation operator sequence $PS = (e^-_{u1a1}, e^-_{u2a2}, e^+_{u1[a1-a2]}, e^+_{u2[a1-a2]})$. Then similar operations are applied to attribute vertices b_1 and b_2. (c) The secrets s_1 and s_2 are perturbed [1] to s_1^* and s_2^* by adding noises to s_1 and s_2 respectively, which could be described by $PS = (s_1^-, s_2^-, s_1^{*+}, s_2{}^*+)$. (d) The G^* is anonymized by anonymity approach *bucketization* [12] via $PS = (e^-_{u1s1}, e^-_{u2s2}, e^+_{u1s2}, e^+_{u2s1})$.

C. Privacy Inference

The privacy violation of user is claimed when (id, s) is correctly inferred with high probability. The graph representations of datasets and assumption of adversary's background knowledge could facilitate the modelling of general privacy inference conducted by attacker in a universal way.

Before going into the detail, it is important to clarify the different roles of anonymity graph G^* and attack graph G_A for secret inference. The associations between users and secrets, specified by path (edge) (u, s), are provided by the anonymity graph G^*. A small number of secret nodes (both explicit secrets such as *salary=20k* and implicit secrets such as *salary=low* in salary hierarchy) that a user node can reach along arbitrary paths indicates a limited diversity or a small error margin of secrets that the user might have. Furthermore, the correlations between secrets and attributes, specified by path (s, u, v), are also represented. The association and correlation makes G^* potentially vulnerable with the presence of attack graph G_A in two aspects. On one hand, the attribute vertices of victims in G_A can help the adversary to de-anonymize u such that the (u, s) association is re-identified as (id, s) which apparently violates the privacy principles. On the other hand, the combination of correlation and BK from G_A could be used by the adversary to further prefer a unique association between u and s rather than others observed from graph G^*. In general, estimating privacy disclosure of G^* is a complex problem (e.g.,$\#P$-complete [12] or NP-hard [10]) when background knowledge are involved. Fig.4 gives a simple case of privacy inferring similar with [9] when graph G^* is seen by adversary with background knowledge G_A. The formal algorithm of privacy leakage estimation under general graph context would be our further work.

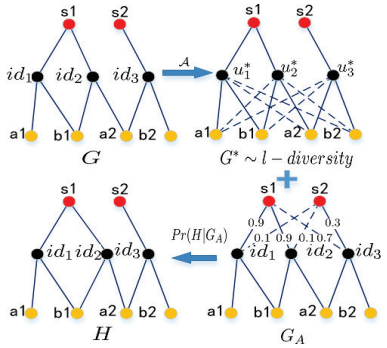

Fig. 4. Example of privacy inferring with background knowledge. Three users (with id_1, id_2, id_3) in original graph G have two attributes (a, b) and one secret (s). The anonymity graph G^* is obtained by anonymizing G with *l-diversity*, where the attributes (a_1, a_2, b_1, b_2) of the three users are generalized by edge addition denotes as dashed lines between user nodes and attribute nodes. The identifications id of users are removed in G^*. If there is no background knowledge, the adversary's belief that user id_3 has secret s_2 is $1/3$. Assume the adversary knows exactly the attributes of all users. The prior belief of secrets of each user are represented by edge weights of attack graph G_A. The values on solid line between node id and s indicate the belief on true secret, and the values on dashed line are the belief on false secret. First, the adversary identifies nodes u_1, u_2, u_3 as potential candidates of id_1, id_2 and id_3 according to the similarity of their attributes. Then based on Bayesian theorem, the adversary tries to re-construct the disclosure graph H by maximum likelihood estimation and infer the posterior belief on secret of each user. In our case, the probability of H is $Pr(H|G_A) = (0.9 \times 0.9 \times 0.3)/((0.9 \times 0.9 \times 0.3) + (0.9 \times 0.1 \times 0.7) + (0.1 \times 0.9 \times 0.7)) = 0.66$, which is the maximum of all the possible disclosure graphs H'. That means, the adversary's belief that user id_3 has secret s_2 increases from prior 0.3 to posterior 0.66. The adversary could successfully reconstruct the exact original graph G with high probability since H is identical to G.

D. Privacy and Utility Metrics

1) Privacy: The example of Fig.4 expresses the privacy criterion of our framework. The privacy of user id_i is said to be protected if the adversary's gain between prior probability Pr_i and posterior probability Pr_i' on the secret s_j is bounded by a reasonable threshold. This is formally stated as:

$$\delta(Pr_i, Pr_i'|G_A, G^*) \leq \epsilon(id_i, s_j) \rightarrow \delta(\omega_{ij}, \omega_{ij}') \leq \epsilon(id_i, s_j) \quad (1)$$

where the function $\delta(x, y) = |x - y|$ is a measurement of privacy gain of adversary, and ω'_{ij} is the edge weight in disclosure graph H inferred by adversary upon seeing the anonymity graph G^*. The threshold ϵ is parameterized by particular (id_i, s_j) so as to permit setting individual protection requirement for different users. This consideration enables *personal* privacy description which essentially brings framework $\mathcal{F}$ the flexibility for various scenarios. A small ϵ means the adversary is merely able to gain trivial privacy information of user. An anonymity graph G^* is said to be privacy preserving against attack capability defined by attack graph G_A if all users satisfy Eq. (1) and denoted as

$$\delta(Pr(S), Pr'(S)|G_A, G^*) \leq \epsilon \tag{2}$$

where S is the set of secrets and $\epsilon = \{\epsilon(id_i, s_j)\}$ is the collection of privacy leakage thresholds for all users. Note that the privacy measurement of graph, the change of adversary's world-view upon seeing the data, is consistent with works such as [3], [9]; however, we extend it under graph context.

2) Utility: While the difference between attack graph G_A and disclosure graph H is used to measure the risk of privacy leakage, the difference between anonymity graph G^* and original data graph G is used as a straightforward selection for utility metric. We use *statistical properties* of graph as basic measurement to quantify the perceived utility by potential applications. The commonly applied statistical properties of dataset include *expectation* and *correlation* which we will redefine on graph. For ease of computation, we define a bipartite graph G_b which contains all the vertices and correlation edges of G. The expectation of attribute vertices with attribute type A_j is defined as $E_{A_j} \triangleq \frac{1}{\sum_i deg(v_i)} \sum_i v_i \times deg(v_i), v_i \in A_j$, Where $deg(v)$ is node degree of vertex v in graph G_b. If A_j is categorical attribute, then E_{A_i} is a vector rather than a scalar where each entry is the ratio of degree of v_i to the summation of degrees of all attribute nodes. The expectation of graph G is then defined as below

$$E_G = (E_{A_1}, \ldots, E_{A_k}) \quad A_j \subset \Sigma \tag{3}$$

The expectation E_{G^*} of G^* can be similarly defined.

Meanwhile, preservation of dependency among multiple attributes is also significant. For computing efficiency, we also define attribute graph G_Σ which is actually a project graph of G. The vertices of G_Σ consists of elements of alphabet $\Sigma = V \cup S$. If there exists a two-hop path from v_i to v_j composed by two correlation edges of G, then a correlation edge between v_i and v_j is constructed in G_Σ. The edge weight ω_{ij} is accordingly defined as the number of distinct paths, which represents the co-occurrence frequency of the two attributes. It can be seen that the correlation edge of G_Σ is actually the weighted projection of correlation paths of G. Note that the semantic edge set of G_Σ is the same as that of G. Given this definition of G_Σ, the correlation of datasets before and after anonymization are directly captured by adjacency matrix of original attribute graph G_Σ and anonymity attribute graph G^*_Σ respectively. The adjacent matrix of G_Σ is therefore called correlation matrix C_G of graph G, that is, $C_G^{|\Sigma|\times|\Sigma|} = \{\omega_{ij}\}$.

Given the definition of expectation and correlation of graph G and G^*, the data utility is quantified by the differences of them before and after anonymization:

$$\begin{cases} U_E = & \delta_E(E_G, E_{G^*}) \\ U_C = & \delta_C(C_G, C_{G^*}) \end{cases} \tag{4}$$

A choice of δ_E is the vector similarity (*e.g.*, cosine similarity), and the function δ_C can be *correlation coefficient* of two matrices. Obviously, $U_E = 1$ and $U_C = 1$ correspond to the maximal utility with zero data distortion, and $U_E = 0$ and $U_C = -1$ correspond to the minimal utility.

IV. Graph Partition for Anonymization

While $\mathcal{F}$ provides a universal framework to model and evaluate the privacy preserving strategy, one of the basic requirements that $\mathcal{F}$ should satisfy is to support various anonymization algorithms efficiently. By endowing the users and attributes with relation and semantic metrics via edges, the privacy protection can be generally considered as graph partition problem of G. Appropriate anonymity operator sequences are applied to each partition rather than to the global graph such that we apply the best anonymization method on each partition. In addition, the graph partition permits flexible subgraph size estimation according to data properties and privacy protection criteria.

A. Problem Formulation

Let cut $\Pi(G) = \{g_i \mid i = 1, 2 \cdots, m\}$ be a partition for graph G. Each subgraph g_i is viewed as an anonymity equivalence class (EC) and the users in g_i are anonymized by the same approach to construct anonymity subgraph g_i^*, that is, $g_i \mapsto g_i^*$. Note that while a user vertex u belongs to only one of the subgraphs, the attribute vertex v or secret vertex s might belong to multiple subgraphs simultaneously. For example, the attribute *Gender:Male* or secret *Location:Chicago* tend to appear in many subgraphs such that the integration of user u and all its neighbours are preserved in g_i. Thus g_i might overlap with each other on their common attribute vertices.

The objective function of our graph partition is naturally posed according to trade-off between privacy guarantee and utility. Intuitively for a subgraph g_i, when each user satisfies the privacy requirement stated by Equ.1, the utility will be high if the similarity of attributes in g_i is maximized (thus it supports better anonymization such as k-anonymity). This intuition leads to a privacy-oriented graph partition algorithm implemented based on attribute graph G_Σ (as defined in section III-D2). A cut $\Pi(G_\Sigma)$ where each subgraph has large summation over weights of both correlation edges (holding more correlation properties) and semantic edges (large weight means high similarity) will produce desirable utility. This statement is equivalent to minimizing the edge weight summation across subgraphs for both edge types, which can be solved as the graph partition with minimum cut. The advantage of performing graph partition on attribute graph G_Σ instead of G mainly arise from its smaller size since user nodes of

G are not contained in G_Σ. This could significantly alleviate the overhead of computation when the size of alphabet $|\Sigma|$ is much less than the user population $|U|$.

B. Graph Partition Algorithm

The problem of graph partition with minimum cut has been extensively studied, among which we adopt the spectrum graph partition paradigm [15]. Assume $cut(V_1, V_2)$ divides the vertices of G_Σ into two subsets V_1 and V_2. Let vector $\mathbf{x} = (x_1, \ldots, x_n)^T$ be the partition assignment of vertices such that $x_i = 1$ when $v_i \in V_1$ and $x_i = -1$ when $v_i \in V_2$. According to Laplacian matrix L of graph, we have

$$x^T L x = \sum_{p=1}^{2} \sum_{(v_i,v_j)\in V_p} (x_i - x_j)^2 \omega_{ij} + \sum_{(v_i,v_j)\in cut} (x_i - x_j)^2 \omega_{ij} \tag{5}$$

Where ω_{ij} is the weight of edge e_{ij}. When v_i, v_j belong to the same subsets, $x_i - x_j = 0$. Thus the first term on right side of Equ.(5) equal to 0. When v_i, v_j belong to different subsets and there is edge between them, $i.e., e_{ij} \in cut(V_1, V_2)$, it will contribute $4\omega_{ij}$ to the summation of last term of Equ.(5) . Thus, the last term can be viewed as the cost of the cut. In addition, to divide graph into two balanced parts with approximately equal amount of nodes, we have the following objective function:

$$\min \Pi(G_\Sigma) = \frac{cut(V_1, V_2)}{|V_1|} + \frac{cut(V_1, V_2)}{|V_2|} \tag{6}$$

The eigenvalues and eigenvectors of normalized Laplacian matrix L provide information to solve the optimal problem of Equ.6. The eigenvector $\mathbf{v_2} = (\nu_1, \ldots, \nu_n)$ (where $\sum_i \nu_i = 0$), corresponding to the second smallest eigenvalue λ_2 of L, is treated to be the cut vector of graph G_Σ. The vertices of G_Σ is then divided into two parts with minimum cut in the way that

$$v_i \in V_1 \text{ if } \nu_i \geq c; \qquad v_i \in V_2 \text{ if } \nu_i < c \tag{7}$$

The constant c is called *splitting value*, which can be chosen by principles including bisection (c is set as the median value of eigenvector $\mathbf{v_2}$, used in experiment), ratio cut, sign cut, or gap cut [16].

So far we have two subsets V_1, V_2 of G_Σ, However, the corresponding subgraphs g_1' and g_2' cannot be directly obtained by solely adding edges between vertices within each vertex subset because all the attributes of a user u are not guaranteed to be assigned to the same subset. To keep the integration of user's attributes, two user subsets U_1 and U_2 are then produced based on V_1 and V_2. For a particular user u, if all her attribute and secret vertices are assigned to the same subset, say V_1 without loss of generality, we simply assign u to U_1. Otherwise if some attributes belong to V_1 but others to V_2, the relative significances of different attributes in terms of data representation are used to decide the assignment of u. Approaches such as PCA (Principle Component Analysis) for categorical variables [17] can be explored to compute the relative significances among attributes. The node u is then assigned to $U_i (i = 1, 2)$ (set of user vertices) such that the corresponding V_i contains the most significant attribute of u. If all the attributes are treated equally significant, u is assigned to U_i such that V_i contains the majority of attributes of u. In addition to above assignment principle, there is also a constraint k posed on the minimum size of U_1 and U_2 in order to satisfy the privacy requirement of equivalent class.

After assigning all users to U_1 or U_2, two new attribute graphs are produced accordingly. The two data graphs generated by users in U_1 and U_2 are projected to attribute graph g_1' and g_2' respectively. As mentioned before, g_1' and g_2' might overlap to each other as users of U_1 and U_2 could have common attributes. The same graph partition process described above are then applied to g_1' and g_2' respectively. If the size of U_i is smaller than $2k$ (for k-anonymity method), the graph partition will stop on g_i' and g_i' acts as an element of partition $\Pi(G_\Sigma)$. Sometimes the g_i' cannot be further divided because all the users of U_i already have identical or similar attributes to each other even the size of g_i' is larger than $2k$. In this case, the g_i' is directly added to $\Pi(G_\Sigma)$. The graph partition will iteratively process until no additional attribute subgraph can be partitioned. At this point, we have the optimal partition $\Pi(G_\Sigma) = \{g_1', \ldots, g_m'\}$. The corresponding $\Pi(G) = \{g_1, \ldots, g_m\}$ can be simply obtained by generating g_i upon g_i' according to graph projection principle.

The graph partition process could be simplified when graph G is a social network with only user nodes. In this case, the graph partition is applied on G instead of G_Σ. Each g_i is a subgraph of G and doesn't overlap with other ones, thus can be obtained directly on V_1 or V_2 instead of U_1 or U_2. For a partition $\{g_1, \ldots, g_m\}$, we cluster the set of g_i according to their structure features(e.g.*node degree* and *edge number*) such that the similar subgraphs are assigned to the same group. Therefore, the outcome partition is a set of subgraph clusters, referred as $\Pi(G) = \{\{g_{1j_1}\}, \{g_{2j_2}\}, \ldots, \{g_{mj_m}\}\}$ where each cluster $\{g_{ij_i}\}$ has at least k subgraphs.

Given the partition $\Pi(G)$, the anonymity operators are applied to each g_i to fulfil privacy criteria. In the case of social network, the anonymity operators are used within each $\{g_{ij_i}\}$. The anonymity graph g_i^* is produced by anonymization function $\mathcal{A}$, i.e.,$g_i^* = \mathcal{A}(g_i)$. The goal of function $\mathcal{A}$ is to ensure that the disclosure graph H meets the privacy requirements imposed by data publisher.

V. Experiment

A. Datasets and Anonymity Approaches

We make use of three different datasets in our experiment to evaluate the efficiency of framework $\mathcal{F}$. First, the *Adult* dataset [3] from US 1994 Census data is selected as case of dataset with a typical table schema. $30,162$ records and 9 attributes (*Age, Workclass, Education, Marital-status, Occupation, Salary, Race, Sex, Native-country*) are selected, where the *salary* and *occupation* are treated as secrets alternatively. The second dataset is social network of *Facebook* which consists of 4039 user nodes and 88234 edges [18], and is divided into 10 ego networks. The associated user profile for each

TABLE I
GRAPH PROPERTIES AND ANONYMITY APPROACHES OF DATASETS

#	Adult	Facebook		Brightkite
		Ego	Profile	
graph	G_Σ	G	G_Σ	G_Σ
record	30162	4039	4039	4491151
vertex	166	4039	396	5044788
edge	9362	174263	28469	9.42292×10^{12}
semantic	2556	174263	2520	9.42291×10^{12}
correlation	6806	n/a	25949	4491151
anonymity approach	Generali zation [20]	pertur bation [5]	swapping [21]	obfuscation [6]

user node includes average 21 attributes in which 6 attributes (*birthday, education degree, education type, education with gender, location, work with*) are selected for our experiment. The final dataset is *Brightkite* [19] with $4,491,151$ check-in records which consists of times and GPS locations. The detail of graph properties of three datasets are shown in Table I: record number, node number, edge number including semantic edges and correlation edges respectively. For clarity, the ego networks and user profile of *Facebook* are shown separately. Note that except the *ego network* of *Facebook*, graph properties are calculated on attribute graph G_Σ instead of G.

To show the capability of framework $\mathcal{F}$ for accommodating various anonymity algorithms, we choose different approaches to anonymize subgraph g_i of three datasets, which are given in the last row of Table I. Note that "we choose" here means we only adopt the data modification approaches and privacy criterion used by the four anonymity strategies, yet the constructions of equivalent class (EC) are all complemented by graph partition rather than those provided by them. For example in social network of *facebook*, we add/delete nodes and edges to make two subgraphs g_i and g_j isomorphic to each other to protect against structure attack, which remains the same with [5]. The g_i and g_j, however, are obtained by graph partition rather than the finding ways of *Potential Anonymization SubGraph* used in [5]. The framework $\mathcal{F}$ is implemented by 64bit Windows 8 system with Core2 T6500@ 2.1GHz CPU and 4G memory.

B. Results

We investigate the efficiency of framework $\mathcal{F}$ from perspectives of privacy, utility, and runtime performance. For simplicity, the attack graph G_A is assumed uniformly distributed over all secrets, thus we only need to be concerned with the privacy leaked by anonymity graph G^*. In this case, the privacy guarantee of G^* is naturally determined by the anonymity approach and is not presented for the limitation of space.

The utility of G^* defined by Eq.(4) for *Adult* dataset is shown in Fig.5. The U_C and U_E are computed under different parameters. In this figure, the utility decreases with the increase of minimal constraint k on user number contained in subgraph g_i since more anonymity operators, necessary to make users indistinguishable, deteriorate the utility. Note that the actual size of g_i has large diversity. For instance, three

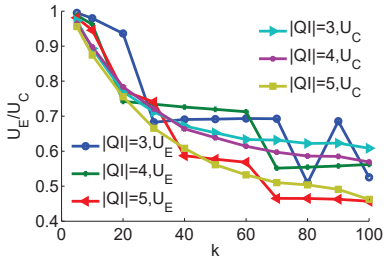

Fig. 5. Graph utility of Adult dataset

TABLE II
UTILITY COMPARISON FOR FRAMEWORK AND FLASH.

		Framework $\mathcal{F}$		flash	
		# g_i	Entropy	# EC	Entropy
Adult salary	k=1=2	153	44983	30	93880
	k=5, l=2	151	45016	30	93880
Adult Occup.	k=1=2	700	30412	15	123947
	k=l=6	73	104868	15	123947
	k=l=10	37	112723	15	123947
	k=l=12	31	119919	8	151819
check-in	k=5	900	49032	12	77884
	k=30	151	65656	12	77884

attributes (*Age,Education,Marital-status*) are selected for 5-anonymity ($|QI| = 3$ and $k = 5$), the maximal number of users in g_i can be as high as 320 in all the 1367 resulting subgrpahs because the 320 users have exactly identical attributes, which can be published directly without any further partition. The utilities generally get worse when more attribute types are involved. That complies with previous observation that higher dimension datasets are harder to be anonymized with a desirable utility.

To comparison, we also anonymize *Adult* using *l-diversity* by *ARX*, which implements *flash* [20] and is a comprehensive tool for data anonymization. The utilities of G^* are re-computed using *Non-uniform entropy*(entropy for short) adopted by *ARX*. The larger the entropy, the worse the utility is. The entropy of *Adult* by *flash* is obtained with three attributes(*Age,Education,Marital-status*) and secrets *salary* or *occupation*. Different parameter combinations (k, l) are tested for comprehensive comparison. Note that the minimal number k of users in g_i is same to the parameter k of *l-diversity*. The results in Table II clearly show that our framework $\mathcal{F}$ outperforms *flash* in terms of utility. The number of subgraphs g_i is far greater than that of ECs obtained by *flash*, which indicates finer granularity of anonymity and hence higher utility. In addition to *Adult*, the *check-in* dataset is also anonymized by *k-anonymity*. The attributes *time* and *location* are treated as QIs in *ARX* to comply with the processing requirement of *flash*. The entropy of framework $\mathcal{F}$ is shown also smaller than that of *flash* on this dataset.

The utility of anonymized graph G^* for *facebook* and *check-in* datasets are summarized in Table III. For social ego network, the number of user nodes in each subgraph of cluster $\{g_{ij_i}\}$ ranges within $[h, 2h-1]$, where $h = 50$ is appropriately selected according to experiment results. The perturbation using node/edge addition/deletion is applied to each subgraph in the same cluster to make them isomorphic and therefore

TABLE III
GRAPH UTILITY OF FACEBOOK AND CHECK-IN

Dataset	U_E	U_C	Runtime(s)
Ego(k=5)	0.808	0.412	164
profile(k=10)	1	0.725	14
check-in(k=10, filter@5)	≈ 1	0.124	891
check-in(k=20, filter@10)	≈ 1	0.085	707
check-in(k=30, filter@15)	≈ 1	0.072	576

k-anonymity is achieved in terms of network structure. The anonymity approach applied for facebook profile is data swapping [21]. The profile of 10 ego networks are independently processed and the average utility are given in Table III. Since the swapping only changes the connection between user nodes and attribute nodes without really change their values, the U_E always equals to 1 which indicates zero distortion of data expectation. The average U_C for 10 profiles is 0.72528, that means the anonymized user profiles are relative similar to the original ones. The check-in data is anonymized by replacing the real time and location with faked ones drawn from the vertices of time and location in the same subgraph according to their probability distribution. The parameter $filter@m$ is used to select the m nearest neighbor vertices as the candidates for replacement. The U_E is approximately 1 and U_C decreases with the increase of user number constrain k, which behaves similar to that of *Adult*. To reduce the computing overhead, the records in check-in data are pre-clustered according to the similarity of times and locations. The average number of records in clusters is 4539 and the utility shown in Table III is the mean over all clusters.

The runtime of framework $\mathcal{F}$ is also shown in Table III, which consists of several stages from data initialization to utility calculation and varies mainly according to graph size and anonymity approaches. For instance, the total runtime to anonymize facebook profile is only 14 seconds which is much smaller than those of other datasets since the small size of its attribute graph G_Σ. To illustrate the effect of each stage, we take *Adult* as an example and plot the runtimes in Fig. 6. It is not surprising that the graph partition takes much more time than other steps and contributes much to the total overhead. Obviously, the computational overhead of runtime decreases with the growing of subgraph size constraint k because large k will usually bring large subgraphs such that fewer iterations in graph partition are needed. The runtime also increases when more attribute types $|QI|$ are involved. As the framework $\mathcal{F}$ typically takes about tens of seconds, the runtime of *flash* is only approximately a few seconds. However, the significant gain on utility still suggests the comprehensive efficiency of framework $\mathcal{F}$.

C. Computation complexity

Without loss of generality, we take k-anonymity as example to illustrate the computation complexity of graph partition algorithm, that is, $O(|V|^3 + |A|^2 \cdot n \cdot \log_2 \frac{n}{k})$. $|V|$ is the total number of vertices in attribute graph G_Σ, $|A|$ is the number of attribute types of G_Σ, n is number of records contained in dataset T, and k is the parameter of k-anonymity. The term $|V|^3$ comes from the matrix decomposition calculation

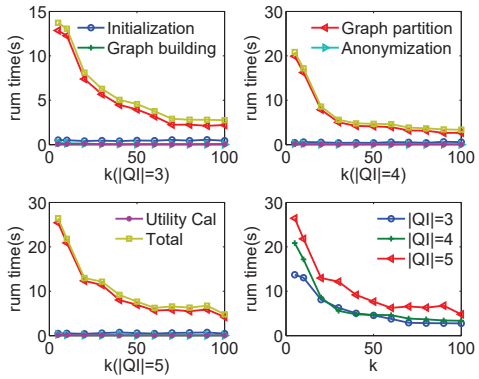

Fig. 6. Runtime of *Adult*. The performances of 5 stages (Initialization, Graph building, Graph partition,Anonymization,Utility calculation) and the total are separately plotted in each subfigure except the bottom right one, where the total runtimes consumed under different number of attribute types QI are given. (The legends for different stages are separately labeled in the three subfigures for the limited space)

for eigenvalues and eigenvector, n/k is the average times of graph partition when graph is evenly partitioned and every subgraph has the size of k, therefore $\log_2(n/k)$ is the average depth of partition iteration. The $O(|V|^3)$ complexity can be further optimized by parallel distributed algorithms when the scale of graph increases.

VI. RELATED WORK

The analysis methodologies of associations among identification, quasi-identifier, and secret are mainly based on the representing data models, where the lattice [2], set [22], metric space [23], and graph [24] are prominently explored.

For dataset which does not appear like graph, the most intuitive choice of graph constructions is from instance level, *i.e.*, a vertex represents a user and an edge represents the relation between users [2], [25]. The attribute level graph is also extensively employed. The vertices are attributes and edges are used to represent the marginal difference [26], privacy indiscrimination property [25], and semantic distances [27]. For dataset with original form of graph, such as social networks, [28] integrated the structural and attribute similarities by inserting a set of attribute vertices to social network. The anonymity strategies for graph mainly includes edge randomization, cluster based generalization, and k-anonymity via edge modification [8]. The *k-automorphism* [24] is a well known instance among them. The *generalization* approaches group vertices and edges into super-nodes and super-edges [8], where the macro-properties consisting of aggregation description of super-nodes are published and the micro-properties in super-nodes are hidden. All these definitions of graph could be viewed as specifications of our framework $\mathcal{F}$.

Among the frameworks for privacy preservation, [29] unified multiple privacy models, where whether a user is in the published dataset or not is viewed as privacy. FRAPP [30] focused on randomized algorithms where each tuple of database is perturbed by replacing it with a randomly chosen tuple according to a predefined probability distribution. The HYDRA [31] was proposed for cross-platform user identity linkage via heterogeneous behavior modeling. Inspired

by *Pufferfish* [32], the *Blowfish privacy* [25] controlled the amount of information disclosed and permitted more utility since not all properties of an individual need to be kept secret. The framework of *versatile publishing* [33] specified the privacy requirements of publishing a microdata table as an arbitrary set of privacy rules. Besides anonymization, the frameworks of privacy preservation were also studied by cryptography-based approaches [34], [35]. Our framework is more general in the sense that it provides a universal way to comprehensively accommodate various anonymity approaches and privacy criteria.

VII. Conclusion

In this paper we present a general graph-based privacy preserving data publication scheme. Most existing privacy protection approaches could be viewed as special cases of our framework. While different privacy threats are enumerated by previous works, the framework provides a potential to derive *unknown privacy threats* under new scenarios or assumptions, which are rarely investigated by previous literature. Clearly, the framework is only the first step to explore the power of graph for privacy preservation. A set of fascinating questions, such as the potential extension to differential privacy [36], [37], and the generation principles of attack graph G_A for various specified background knowledges, need to be addressed to fully materialize the power of such graph-based framework.

VIII. Acknowledgment

This work is supported by National Natural Science Foundation of China under Grant No. 61302077, Grant No. 61520106007, 863 Project under Grant No.2014AA01A706, and NSF ECCS-1247944, NSF ECCS-1343306, NSF CMMI 1436786, NSF CNS 1526638

References

[1] B. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Computing Surveys (CSUR)*, vol. 42, no. 4, p. 14, 2010.

[2] V. Ciriani, S. D. C. Di Vimercati, S. Foresti, and P. Samarati, "K-anonymity," *Advances in Information Security*, 2007.

[3] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM TKDD*, vol. 1, no. 1, p. 3, 2007.

[4] D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *ACM PODS*. ACM, 2001, pp. 247–255.

[5] J. Cheng, A. W.-c. Fu, and J. Liu, "K-isomorphism: privacy preserving network publication against structural attacks," in *SIGMOD*. ACM, 2010, pp. 459–470.

[6] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Protecting location privacy: optimal strategy against localization attacks," in *CCS*. ACM, 2012, pp. 617–627.

[7] T. Dalenius, "Finding a needle in a haystack-or identifying anonymous census record," *Journal of official statistics*, vol. 2, no. 3, pp. 329–336, 1986.

[8] X. Wu, X. Ying, K. Liu, and L. Chen, "A survey of privacy-preservation of graphs and social networks," in *Managing and mining graph data*. Springer, 2010, pp. 421–453.

[9] T. Li, N. Li, and J. Zhang, "Modeling and integrating background knowledge in data anonymization," in *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*. IEEE, 2009, pp. 6–17.

[10] B.-C. Chen, K. LeFevre, and R. Ramakrishnan, "Privacy skyline: Privacy with multidimensional adversarial knowledge," in *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, 2007, pp. 770–781.

[11] "Open source project: Arx," http://arx.deidentifier.org/.

[12] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Y. Halpern, "Worst-case background knowledge for privacy-preserving data publishing," in *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*. IEEE, 2007, pp. 126–135.

[13] J. Qian, X.-Y. Li, C. Zhang, and l. Chen, "De-anonymizing social networks and inferring private attributes using knowledge graphs," in *INFOCOM*. IEEE, 2016.

[14] W. Du, Z. Teng, and Z. Zhu, "Privacy-maxent: integrating background knowledge in privacy quantification," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 2008, pp. 459–472.

[15] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *SIGKDD*. ACM, 2001, pp. 269–274.

[16] D. A. Spielman and S.-H. Teng, "Spectral partitioning works: Planar graphs and finite element meshes," *Linear Algebra and its Applications*, vol. 421, no. 2, pp. 284–305, 2007.

[17] H. Niitsuma and T. Okada, "Covariance and pca for categorical variables," in *Advances in Knowledge Discovery and Data Mining*. Springer, 2005, pp. 523–528.

[18] J. Leskovec and J. J. Mcauley, "Learning to discover social circles in ego networks," in *Advances in neural information processing systems*, 2012, pp. 539–547.

[19] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *SIGKDD*. ACM, 2011, pp. 1082–1090.

[20] F. Kohlmayer, F. Prasser, C. Eckert, A. Kemper, and K. A. Kuhn, "Flash: efficient, stable and optimal k-anonymity," in *PASSAT and SocialCom*. IEEE, 2012, pp. 708–717.

[21] J. Domingo-Ferrer and V. Torra, "Theory and practical applications for statistical agencies," *North-Holland: Amsterdam*, pp. 113–134, 2002.

[22] C. Yao, X. S. Wang, and S. Jajodia, "Checking for k-anonymity violation by views," in *VLDB*. VLDB Endowment, 2005, pp. 910–921.

[23] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu, "Achieving anonymity via clustering," in *ACM Symposium on Principles of database systems*. ACM, 2006, pp. 153–162.

[24] L. Zou, L. Chen, and M. T. Özsu, "K-automorphism: A general framework for privacy preserving network publication," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 946–957, 2009.

[25] X. He, A. Machanavajjhala, and B. Ding, "Blowfish privacy: Tuning privacy-utility trade-offs using policies," in *SIGMOD*. ACM, 2014, pp. 1447–1458.

[26] D. Kifer and J. Gehrke, "Injecting utility into anonymized datasets," in *SIGMOD*. ACM, 2006, pp. 217–228.

[27] M. Kroll, "A graph theoretic linkage attack on microdata in a metric space," *arXiv preprint arXiv:1402.3198*, 2014.

[28] Y. Zhou, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 718–729, 2009.

[29] N. Li, W. H. Qardaji, D. Su, and Y. W. andWeining Yang, "Membership privacy: a unifying framework for privacy definitions," in *ACM CCS*.

[30] S. Agrawal and J. R. Haritsa, "A framework for high-accuracy privacy-preserving mining," in *ICDE*. IEEE, 2005, pp. 193–204.

[31] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan, "Hydra: Large-scale social identity linkage via heterogeneous behavior modeling," in *SIGMOD*. ACM, 2014, pp. 51–62.

[32] D. Kifer and A. Machanavajjhala, "Pufferfish: A framework for mathematical privacy definitions," *ACM Transactions on Database Systems (TODS)*, vol. 39, no. 1, p. 3, 2014.

[33] X. Jin, M. Zhang, N. Zhang, and G. Das, "Versatile publishing for privacy preservation," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 353–362.

[34] T. Jung, X.-Y. Li, and J. Han, "A framework for optimization in big data: Privacy-preserving multi-agent greedy algorithm," in *Big Data Computing and Communications*. Springer, 2015, pp. 88–102.

[35] L. Zhang, T. Jung, C. Liu, X. Ding, X.-Y. Li, and Y. Liu, "Pop: Privacy-preserving outsourced photo sharing and searching for mobile devices," in *ICDCS, 2015 Proceedings IEEE*. IEEE, 2015, pp. 308–317.

[36] C. Dwork, "Differential privacy," in *ICALP*, 2006, pp. 1–12.

[37] J. Zhao, T. Jung, Y. Wang, and X. Li, "Achieving differential privacy of data disclosure in the smart grid," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 504–512.