Chapter 1

A SURVEY OF ALGORITHMS FOR PRIVACY-PRESERVATION OF GRAPHS AND SOCIAL NETWORKS

Xintao Wu
University of North Carolina at Charlotte
xwu@uncc.edu

Xiaowei Ying
University of North Carolina at Charlotte
xying@uncc.edu

Kun Liu
Yahoo! Labs
kun@yahoo-inc.com

Lei Chen
Hong Kong University of Science and Technology
leichen@cs.ust.hk

Abstract Social networks have received dramatic interest in research and development. In this chapter, we survey the very recent research development on privacy-preserving publishing of graphs and social network data. We categorize the state-of-the-art anonymization methods on simple graphs in three main categories: K-anonymity based privacy preservation via edge modification, probabilistic privacy preservation via edge randomization, and privacy preservation via generalization. We then review anonymization methods on rich graphs. We finally discuss challenges and propose new research directions in this area.

Keywords: Anonymization, Randomization, Generalization, Privacy Disclosure, Social Networks

1. Introduction

Social networks are of significant importance in various application domains such as marketing, psychology, epidemiology and homeland security. The management and analysis of these networks have attracted increasing interests in the sociology, database, data mining and theory communities. Most previous studies are focused on revealing interesting properties of networks and discovering efficient and effective analysis methods [5–7, 14, 15, 23–25, 27, 36–40].

Social networks often contain some private attribute information about individuals as well as their sensitive relationships. Many applications of social networks such as anonymous Web browsing require identity and/or relationship anonymity due to the sensitive, stigmatizing, or confidential nature of user identities and their behaviors. The privacy concerns associated with data analysis over social networks have incurred the recent research. In particular, privacy disclosure risks arise when the data owner wants to publish or share the social network data with another party for research or business-related applications. Privacy-preserving social network publishing techniques are usually adopted to protect privacy through masking, modifying and/or generalizing the original data while without sacrificing much data utility. In this chapter, we provide a detailed survey of the *very recent* work on this topic in an effort to allow readers to observe common themes and future directions.

1.1 Privacy in Publishing Social Networks

In a social network, nodes usually correspond to individuals or other social entities, and an edge corresponds to the relationship between two entities. Each entity can have a number of attributes, such as age, gender, income, and a unique identifier. One common practice to protect privacy is to publish a naive node-anonymized version of the network, e.g., by replacing the identifying information of the nodes with random IDs. While the naive node-anonymized network permits useful analysis, as first pointed out in [4, 20], this simple technique does not guarantee privacy since adversaries may re-identify a target individual from the anonymized graph by exploiting some known structural information of his neighborhood.

The privacy breaches in social networks can be grouped to three categories: *identity disclosure*, *link disclosure*, and *attribute disclosure*. The identity disclosure corresponds to the scenario where the identity of an individual who is associated with a node is revealed. The link disclosure corresponds to the scenario where the sensitive relationship between two

individuals is disclosed. The attribute disclosure denotes the sensitive data associated with each node is compromised. Compared with existing anonymization and perturbation techniques of tabular data, it is more challenging to design effective anonymization techniques for social network data because of difficulties in modeling background knowledge and quantifying information loss.

1.2 Background Knowledge

Adversaries usually rely on background knowledge to de-anonymize nodes and learn the link relations between de-anonymized individuals from the released anonymized graph. The assumptions of the adversary's background knowledge play a critical role in modeling privacy attacks and developing methods to protect privacy in social network data. In [51], Zhou et al. listed several types of background knowledge: attributes of vertices, specific link relationships between some target individuals, vertex degrees, neighborhoods of some target individuals, embedded subgraphs, and graph metrics (e.g., betweenness, closeness, centrality).

For simple graphs in which nodes are not associated with attributes and links are unlabeled, adversaries only have structural background knowledge in their attacks (e.g., vertex degrees, neighborhoods, embedded subgraphs, graph metrics). For example, Liu and Terzi [31] considered vertex degrees as background knowledge of the adversaries to breach the privacy of target individuals, the authors of [19, 20, 50] used neighborhood structural information of some target individuals, the authors of [4, 52] proposed the use of embedded subgraphs, and Ying and Wu [47] exploited the topological similarity/distance to breach the link privacy.

For rich graphs in which nodes are associated with various attributes and links may have different types of relationships, it is imperative to study the impact on privacy disclosures when adversaries combine attributes and structural information together in their attacks. Re-identification with attribute knowledge of individuals has been well-studied and resiting techniques have been developed for tabular data (see, e.g., the survey book [1]). However, applying those techniques directly on network data erases inherent graph structural properties. The authors, in [8, 9, 11, 49], investigated anonymization techniques for different types of rich graphs against complex background knowledge.

As pointed out in two earlier surveys [30, 51], it is very challenging to model all types of background knowledge of adversaries and quantify

their impacts on privacy breaches in the scenario of publishing social networks with privacy preservation.

1.3 Utility Preservation

An important goal of publishing social network data is to permit useful analysis tasks. Different analysis tasks may expect different utility properties to be preserved. So far, three types of utility have been considered.

- Graph topological properties. One of the most important applications of social network data is for analyzing graph properties. To understand and utilize the information in a network, researches have developed various measures to indicate the structure and characteristics of the network from different perspectives [12]. Properties including degree sequences, shortest connecting paths, and clustering coefficients are addressed in [19, 20, 31, 45, 46, 50].

- Graph spectral properties. The spectrum of a graph is usually defined as the set of eigenvalues of the graph's adjacency matrix or other derived matrices. The graph spectrum has close relations with many graph characteristics and can provide global measures for some network properties [36]. Spectral properties are adopted to preserve utility of randomized graphs in [45, 46].

- Aggregate network queries. An aggregate network query calculates the aggregate on some paths or subgraphs satisfying some query conditions. One example is that the average distance from a medical doctor vertex to a teacher vertex in a network. In [8, 11, 50, 52], the authors considered the accuracy of answering aggregate network queries as the measure of utility preservation.

In general, it is very challenging to quantify the information loss in anonymizing social networks. For tabular data, since each tuple is usually assumed to be independent, we can measure the information loss of the anonymized table using the sum of the information loss of each individual tuple. However, for social network data, the information loss due to the graph structure change should also be taken into account in addition to the information loss associated with node attribute changes. In [52], Zou et al. used the number of modified edges between the original graph and the released one to quantify information loss due to structure change. The rationale of using anonymization cost to measure the information loss is that a lower anonymization cost indicates that fewer changes have been made to the original graph.

1.4 Anonymization Approaches

Similar to the design of anonymization methods for tabular data, the design of anonymization methods also need take into account the attacking models and the utility of the data. We categorize the state-of-the-art anonymization methods on simple network data into three categories as follows.

- K-anonymity privacy preservation via edge modification. This approach modifies graph structure via a sequence of edge deletions and additions such that each node in the modified graph is indistinguishable with at least $K-1$ other nodes in terms of some types of structural patterns.

- Edge randomization. This approach modifies graph structure by randomly adding/deleting edges or switching edges. It protects against re-identification in a probabilistic manner.

- Clustering-based generalization. This approach clusters nodes and edges into groups and anonymizes a subgraph into a super-node. The details about individuals are hidden.

The above anonymization approaches have been shown as a necessity in addition to naive anonymization to preserve privacy in publishing social network data.

In the following, we first focus on *simple graphs* in Section 2 to 5. Specifically, we revisit existing attacks on naive anonymized graphs in Section 2, K-anonymity approaches via edge modification in Section 3, edge randomization approaches in Section 4, and clustering-based generalization approaches in Section 5 respectively. We then survey the recent development of anonymization techniques for *rich graphs* in Section 6. Section 7 is dedicated to other privacy issues in online social networks in addition to those on publishing social network data. We give conclusions and point out future directions in Section 8.

1.5 Notations

A network $G(V, E)$ is a set of n nodes connected by a set of m links, where V denotes the set of nodes and $E \subseteq V \times V$ is the set of links. The network considered here is binary, symmetric, and without self-loops. $A = (a_{ij})_{n\times n}$ is the adjacency matrix of G: $a_{ij} = 1$ if node i and j are connected and $a_{ij} = 0$ otherwise. The degree of node i, d_i, is the number of the nodes connected to node i, i.e., $d_i = \sum_j a_{ij}$, and $\boldsymbol{d} = \{d_1, \ldots, d_n\}$ denotes the degree sequence. The released graph after perturbation is

denoted by $\widetilde{G}(\widetilde{V}, \widetilde{E})$. $\widetilde{A} = (\tilde{a}_{ij})_{n \times n}$ is the adjacency matrix of $\widetilde{G}$, and $\tilde{d}_i$ and $\tilde{\boldsymbol{d}}$ are the degree and degree sequence of $\widetilde{G}$ respectively.

Note that, for ease of presentation, we use the following pairs of terms interchangeably: "graph" and "network", "node" and "vertex", "edge" and "link", "entity" and "individual", "attacker" and "adversary".

2. Privacy Attacks on Naive Anonymized Networks

The practice of naive anonymization replaces the personally identifying information associated with each node with a random ID. However, an adversary can potentially combine external knowledge with the observed graph structure to compromise privacy, de-anonymize nodes, and learn the existence of sensitive relationships between explicitly de-anonymized individuals.

2.1 Active Attacks and Passive Attacks

In [24], Backstrom et al. presented two different types of attacks on anonymized social networks.

- **Active attacks.** An adversary chooses an arbitrary set of target individuals, creates a small number of new user accounts with edges to these target individuals, and establishes a highly distinguishable pattern of links among the new accounts. The adversary can then efficiently find these new accounts together with the target individuals in the released anonymized network.
- **Passive attacks.** An adversary does not create any new nodes or edges. Instead, he simply constructs a coalition, tries to identify the subgraph of this coalition in the released network, and compromises the privacy of neighboring nodes as well as edges among them.

The *active attack* is based on the uniqueness of small subgraphs embedded in the network. The constructed subgraph H by the adversary needs to satisfy the following three properties in order to make the *active attack* succeed:

- There is no other subgraph S in G such that S and H are isomorphic.
- H is uniquely and efficiently identifiable regardless of G.
- The subgraph H has no non-trivial automorphisms.

It has been shown theoretically that a randomly generated subgraph H formed by $O(\sqrt{\log n})$ nodes can compromise the privacy of arbitrarily target nodes with high probability for any network. The *passive attack* is based on the observation that most nodes in real social network data already belong to a small uniquely identifiable subgraph. A coalition X of size k is initiated by one adversary who recruits $k-1$ of his neighbors to join the coalition. It assumes that the users in the coalition know both the edges amongst themselves (i.e., the internal structure of H) and the names of their neighbors outside X. Since the structure of H is not randomly generated, there is no guarantee that it can be uniquely identified. The primary disadvantage of the *passive attack* in practice, compared to the *active attack*, is that it does not allow one to compromise the privacy of arbitrary users. The adversaries can adopt a hybrid *semi-passive* attack: they create no new accounts, but simply create a few additional out-links to target users before the anonymized network is released. We refer readers to [24] for more details on theoretical results and empirical evaluations on a real social network with 4.4 million nodes and 77 million edges extracted from LiveJoural.com.

2.2 Structural Queries

In [19], Hay et al. studied three types of background knowledge to be used by adversaries to attack naively-anonymized networks. They modeled adversaries' external information as the access to a source that provides answers to a *restricted knowledge query* Q about a single target node in the original graph. Specifically, background knowledge of adversaries is modeled using the following three types of queries.

- **Vertex refinement queries.** These queries describe the local structure of the graph around a node in an iterative refinement way. The weakest knowledge query, $\mathcal{H}_0(x)$, simply returns the label of the node x; $\mathcal{H}_1(x)$ returns the degree of x; $\mathcal{H}_2(x)$ returns the multiset of each neighbors' degree, and $\mathcal{H}_i(x)$ can be recursively defined as:

$$\mathcal{H}_i(x) = \{\mathcal{H}_{i-1}(z_1), \mathcal{H}_{i-1}(z_2), \cdots, \mathcal{H}_{i-1}(z_{d_x})\}$$

  where $z_1, \cdots, z_{d_x}$ are the nodes adjacent to x.

- **Subgraph queries.** These queries can assert the existence of a subgraph around the target node. The descriptive power of a query is measured by counting the number of edges in the described subgraph. The adversary is capable of gathering some fixed number of edges focused around the target x. By exploring the neighborhood

of x, the adversary learns the existence of a subgraph around x representing partial information about the structure around x.

- **Hub fingerprint queries.** A hub is a node in a network with high degree and high betweenness centrality. A hub fingerprint for a target node x, $\mathcal{F}_i(x)$, is a description of the node's connections to a set of designated hubs in the network where the subscript i places a limit on the maximum distance of observable hub connections.

The above queries represent a range of structural information that may be available to adversaries, including complete and partial descriptions of node's local neighborhoods, and node's connections to hubs in the network.

Vertex refinement queries provide complete information about node degree while a subgraph query can never express $\mathcal{H}_i$ knowledge because subgraph queries are existential and cannot assert exact degree constraints or the absence of edges in a graph. The semantics of subgraph queries seem to model realistic adversary capabilities more accurately. It is usually difficult for an adversary to acquire the complete detailed structural description of higher-order vertex refinement queries.

2.3 Other Attacks

In [34], Narayanan and Shmatikov assumed that the adversary has two types of background knowledge: aggregate auxiliary information and individual auxiliary information. The aggregate auxiliary information includes an auxiliary graph $G_{\mathsf{aux}}(V_{\mathsf{aux}}, E_{\mathsf{aux}})$ whose members overlap with the anonymized target graph and a set of probability distributions defined on attributes of nodes and edges. These distributions represent the adversary's (imperfect) knowledge of the corresponding attribute values. The individual auxiliary information is the detailed information about a very small number of individuals (called *seeds*) in both the auxiliary graph and the target graph.

After re-identifying the seeds in target graph, the adversaries immediately get a set of de-anonymized nodes. Then, by comparing the neighborhoods of the de-anonymized nodes in the target graph with the auxiliary graph, the adversary can gradually enlarge the set of de-anonymized nodes. During this *propagation* process, known information such as probability distributions and mappings are updated repeatedly to reduce the error. The authors showed that even some edge addition and deletion are applied independently to the released graph and the auxiliary graph, their de-anonymizing algorithm can correctly re-identify a large number of nodes in the released graph.

To protect against these attacks, researchers have developed many different privacy models and graph anonymization methods. Next, we will provide a detailed survey on these techniques.

3. *K*-Anonymity Privacy Preservation via Edge Modification

The adversary aims to locate the vertex in the network that corresponds to the target individual by analyzing topological features of the vertex based on his background knowledge about the individual. Whether individuals can be re-identified depends on the descriptive power of the adversary's background knowledge and the structural similarity of nodes. To quantify the privacy breach, Hey et al. [19] proposed a general model for social networks as follows:

DEFINITION 1 ***K*-candidate anonymity.** *A node x is K-candidate anonymous with respect to a structure query Q if there exist at least $K-1$ other nodes in the graph that match query Q. In other words, $|cand_Q(x)| \geq K$ where $cand_Q(x) = \{y \in V | Q(y) = Q(x)\}$. A graph satisfies K-candidate anonymity with respect to Q if all the nodes are K-candidate anonymous with respect to Q.*

Three types of queries (vertex refinement queries, subgraph queries, and hub fingerprint queries) were presented and evaluated on the naive anonymized graphs. In [20], Hay et al. studied an edge randomization technique that modifies the graph via a sequence of random edge deletions followed by edge additions. In [19] Hay et al. presented a generalization technique that groups nodes into super-nodes and edges into super-edges to satisfy the K-anonymity. We will introduce their techniques in Section 4.1 and 5 in details respectively.

Several methods have been investigated to prevent node re-identification based on the K-anonymity concept. These methods differ in the types of the structural background knowledge that an adversary may use. In [31], Liu and Terzi assumed that the adversary knows only the degree of the node of a target individual. In [50], Zhou and Pei assumed one specific subgraph constructed by the immediate neighbors of a target node is known. In [52], Zou et al. considered all possible structural information around the target and proposed K-automorphism to guarantee privacy under any structural attack.

3.1 *K*-Degree Generalization

In [31], Liu and Terzi pointed out that the degree sequences of real-world graphs are highly skewed, and it is usually easy for adversaries to

collect the degree information of a target individual. They investigated how to modify a graph via a set of edge addition (and/or deletion) operations in order to construct a new K-degree anonymous graph, in which every node has the same degree with at least $K-1$ other nodes. The authors imposed a requirement that the minimum number of edge-modifications is made in order to preserve the utility. The K-degree anonymity property prevents the re-identification of individuals by the adversaries with prior knowledge on the number of social relationships of certain people (i.e., vertex background knowledge).

DEFINITION 2 ***K*-degree anonymity.** *A graph $G(V,E)$ is K-degree anonymous if every node $u \in V$ has the same degree with at least $K-1$ other nodes.*

PROBLEM 1 *Given a graph $G(V,E)$, construct a new graph $\widetilde{G}(\widetilde{V},\widetilde{E})$ via a set of edge-addition operations such that 1) $\widetilde{G}$ is K-degree anonymous; 2)$V=\widetilde{V}$; and 3) $\widetilde{E}\cap E = E$.*

The proposed algorithm is outlined below.

1 Starting from the degree sequence $\boldsymbol{d}$ of the original graph $G(V,E)$, construct a new degree sequence $\tilde{\boldsymbol{d}}$ that is K-anonymous and the L_1 distance, $\|\tilde{\boldsymbol{d}}-\boldsymbol{d}\|_1$ is minimized.

2 Construct a new graph $\widetilde{G}(\widetilde{V},\widetilde{E})$ such that $\boldsymbol{d}_{\widetilde{G}}=\tilde{\boldsymbol{d}}$, $\widetilde{V}=V$, and $\widetilde{E}=E$ (or $\widetilde{E}\cap E \approx E$ in the relaxed version).

The first step is solved by a linear-time dynamic programming algorithm while the second step is based on a set of graph-construction algorithms given a degree sequence. The authors also extended their algorithms to allow for simultaneous edge additions and deletions. Their empirical evaluations showed that the proposed algorithms can effectively preserve the graph utility (in terms of topological features) while satisfying the K-degree anonymity.

3.2 *K*-Neighborhood Anonymity

In [50], Zhou and Pei assumed that the adversary knows subgraph constructed by the immediate neighbors of a target node. The proposed greedy graph-modification algorithm generalizes node labels and inserts edges until each neighborhood is indistinguishable to at least $K-1$ others.

DEFINITION 3 ***K*-neighborhood anonymity.** *A node u is K-neighborhood anonymous if there exist at least $K-1$ other nodes $v_1,\ldots,v_{K-1} \in V$*

such that the subgraph constructed by the immediate neighbors of each node $v_1, \cdots, v_{K-1}$ is isomorphic to the subgraph constructed by the immediate neighbors of u. A graph satisfies K-neighborhood anonymity if all the nodes are K-neighborhood anonymous.

The definition can be extended from the immediate neighbor to the d-neighbors ($d > 1$) of the target vertex, i.e., the vertices within distance d to the target vertex in the network.

PROBLEM 2 *Given a graph $G(V, E)$, construct a new graph $\widetilde{G}(\widetilde{V}, \widetilde{E})$ satisfying the following conditions: 1) $\widetilde{G}$ is K-neighborhood anonymous; 2)$V = \widetilde{V}$; 3) $\widetilde{E} \cap E = E$; and 4) $\widetilde{G}$ can be used to answer aggregate network queries as accurately as possible.*

The simple case of constructing a K-neighborhood anonymous graph satisfying condition 1-3) was shown as *NP*-hard [50]. The proposed algorithm is outlined below.

1 Extract the neighborhoods of all vertices in the network. A *neighborhood component coding* technique, which can represent the neighborhoods in a concise way, is used to facilitate the comparisons among neighborhoods of different vertices including the isomorphism tests.

2 Organize vertices into groups and anonymize the neighborhoods of vertices in the same group until the graph satisfies K-anonymity. A heuristic of starting with vertices with high degrees is adopted since these vertices are more likely to be vulnerable to structural attacks.

In [50], Zhou and Pei studied social networks with vertex attributes information in addition to the unlabeled network topology. The vertex attributes form a hierarchy. Hence, there are two ways to anonymize the neighborhoods of vertices: generalizing vertex labels and adding edges. In terms of utility, it focuses on using anonymized social networks to answer aggregate network queries.

3.3 *K*-Automorphism Anonymity

Zou et al. in [52] adopted a more general assumption: the adversary can know any subgraph around a certain individual α. If such a subgraph can be identified in the anonymized graph with high probability, user α has a high identity disclosure risk. The authors aimed to construct a graph $\widetilde{G}$ so that for any subgraph $X \subset G$, $\widetilde{G}$ contains at least K

subgraphs isomorphic to X. We first give some definitions introduced in [52]:

DEFINITION 4 ***Graph isomorphism and automorphism.*** *Given two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$, G_1 is isomorphic to G_2 if there exists a bijective function $f : V_1 \to V_2$ such that for any two nodes $u, v \in V_1$, $(u, v) \in E_1$ if and only if $(f(u), f(v)) \in E_2$. If G_1 is isomorphic to itself under function f, G_1 is an automorphic graph, and f is called an automorphic function of G_1.*

DEFINITION 5 ***K-automorphic graph.*** *Graph G is a K-automorphic graph if 1) there exist $K-1$ non-trivial automorphic functions of G, $f_1, \ldots, f_{K-1}$; and 2) for any node u, $f_i(u) \neq f_j(u)$ $(i \neq j)$.*

If the released graph $\widetilde{G}$ is a K-automorphic graph, when the adversary tries to re-identify node u through a subgraph, he will always get at least K different subgraphs in $\widetilde{G}$ that match his subgraph query. With the second condition in Definition 5, it is guaranteed that the probability of a successful re-identification is no more than $\frac{1}{K}$. The second condition in Definition 5 is necessary to guarantee the privacy safety. If it is violated, the worst case is that for a certain node u and any $i = 1, 2, \ldots, K-1$, $f_i(u) \equiv u$, and the adversary can then successfully re-identify node u in $\widetilde{G}$. For example, consider a l-asteroid graph in which a central node is connected by l satellite nodes and the l satellite nodes are not connected to each other. This l-asteroid graph has at least l automorphic functions. However the central node is always mapped to itself by any automorphic function. Condition 2 prevents such cases from happening in the released graph $\widetilde{G}$. The authors then considered the following problem:

PROBLEM 3 *Given the original graph G, construct graph $\widetilde{G}$ such that $E \subseteq \widetilde{E}$ and $\widetilde{G}$ is a K-automorphic graph.*

The following steps briefly show the framework of their algorithm:

1 Partition graph G into several groups of subgraphs $\{U_i\}$, and each group U_i contains $K_i \geq K$ subgraphs $\{P_{i1}, P_{i2}, \ldots, P_{iK_i}\}$ where any two subgraphs do not share a node or edge.

2 For each U_i, make $P_{ij} \in U_i$ isomorphic to each other by adding edges. Then, there exists function $f^{(i)}_{s,t}(\cdot)$ under which P_{is} is isomorphic to P_{it}.

3 For each edge (u, v) across two subgraphs, i.e. $u \in P_{ij}$ and $v \in P_{st}$ $(P_{ij} \neq P_{st})$, add edge $\left(f^{(i)}_{j,\pi_j(r)}(u), f^{(s)}_{t,\pi_t(r)}(v)\right)$, where $\pi_j(r) = (j+r) \mod K$, $r = 1, 2, \ldots, K-1$.

After the modification, for any node u, suppose $u \in P_{ij}$, define $f_r(\cdot)$ as $f_r(u) = f^{(i)}_{j,\pi_j(r)}(u)$, $r = 1, \ldots, K-1$. Then, $f_r(u)$, $r = 1, \ldots, K-1$, are $K-1$ non-trivial automorphic functions of $\widetilde{G}$, and for any $s \neq t$, $f_s(u) \neq f_t(u)$, which guarantees the K-automorphism.

To better preserve the utility, the authors expected that the above algorithm introduces the minimal number of fake edges, which implies that subgraphs within one group U_i should be very similar to each other (so that Step 2 only introduces a small number of edges), and there are few edges across different subgraphs (so that Step 3 will not add many edges). This depends on how the graph is partitioned. If G is partitioned into fewer subgraphs, there are fewer crossing edges to be added. However, fewer subgraphs imply that the size of each subgraph is large, and more edges within each subgraph need to be added in Step 2. The authors proved that to find the optimal solution is *NP*-complete, and they proposed a greedy algorithm to achieve the goal.

In addition to proposing the K-automorphism idea to protect the graph under any structural attack, the authors also studied an interesting problem with respect to privacy protection over dynamic releases of graphs. Specially, the requirements of social network analysis and mining demand releasing the network data from time to time in order to capture the evolution trends of these data. The existing privacy-preserving methods only consider the privacy protection in "one-time" release. The adversary can easily collect the multiple releases and identify the target through comparing the difference among these releases. Zou et al. [52] extended the solution of K-automorphism by publishing the vertex ID set instead of single vertex ID for the high risk nodes.

4. Privacy Preservation via Randomization

Besides K-anonymity approaches, randomization is another widely adopted strategy for privacy-preserving data analysis. Additive noise based randomization approaches have been well investigated in privacy-preserving data mining for numerical data (e.g., [2, 3]). For social networks, two edge-based randomization strategies have been commonly adopted.

- *Rand Add/Del*: randomly add k false edges followed by deleting k true edges. This strategy preserves the total number of edges in the original graph.

- *Rand Switch*: randomly switch a pair of existing edges (t, w) and (u, v) (satisfying edge (t, v) and edge (u, w) do not exist in G) to

(t, v) and (u, w), and repeat this process for k times. This strategy preserves the degree of each vertex.

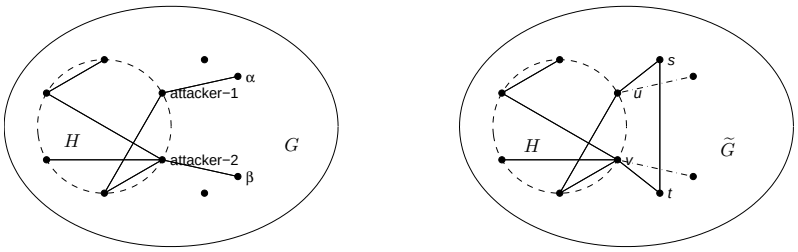
The process of randomization and the randomization parameter k are assumed to be published along with the released graph. By using adjacency matrix, the edge randomization process can be expressed in the matrix form $\widetilde{A} = A + E$, where E is the perturbation matrix: $E(i, j) = E(j, i) = 1$ if edge (i, j) is added, $E(i, j) = E(j, i) = -1$ if edge (i, j) is deleted, and 0 otherwise. Naturally, edge randomization can also be considered as an additive-noise perturbation. After the randomization, the randomized graph is expected to be different from the original one. As a result, the node identities as well as the true sensitive or confidential relationship between two nodes are protected.

In this section, we first discuss why randomized graphs are resilient to structural attacks and how well randomization approaches can protect node identity in Section 4.1. Notice that the randomization approaches protect against re-identification in a probabilistic manner, and hence they cannot guarantee that the randomized graphs satisfy K-anonymity strictly.

There exist some scenarios that node identities (and even entity attributes) are not confidential but sensitive links between target individuals are confidential and should be protected. For example, in a transaction network, an edge denoting a financial transaction between two individuals is considered confidential while nodes corresponding to individual accounts is non-confidential. In these cases, data owners can release the edge randomized graph without removing node annotations. We study how well the randomization approaches protect sensitive links in Section 4.2.

An advantage of randomization is that many features could be accurately reconstructed from the released randomized graph. However, distribution reconstruction methods (e.g., [2, 3]) designed for numerical data could not be applied on network data directly since the randomization mechanism in social networks (based on the positions of randomly chosen edges) is much different from the additive noise randomization (based on random values for all entries). We give an overview of low rank approximation based reconstruction methods in Section 4.3.

Edge randomization may significantly affect the utility of the released randomized graph. We survey some randomization strategies that can preserve structural properties in Section 4.4.



(a) the original graph (b) the released graph

Figure 1.1. Resilient to subgraph attacks

4.1 Resilience to Structural Attacks

Recall that in both *active attacks* and *passive attacks* [4], the adversary needs to construct a highly distinguishable subgraph H with edges to a set of target nodes, and then to re-identify the subgraph and consequently the targets in the released anonymized network. As shown in Figure 1.1(a), attackers form an subgraph H in the original graph G, and attacker 1 and 2 send links to the target individuals α and β. After randomization using either *Rand Add/Del* or *Rand Switch*, the structure of subgraph H as well G is changed. The re-identifiability of the subgraph H from the randomized released graph $\widetilde{G}$ may significantly decrease when the magnitude of perturbation is medium or large. Even if the subgraph H can still be distinguished, as shown in Figure 1.1(b), link (u, s) and (v, t) in $\widetilde{G}$ can be false links. Hence node s and t do not correspond to target individuals α and β. Furthermore, even individuals α and β have been identified, the observed link between α and β can still be a false link. Hence, the link privacy can still be protected. In summary, it is more difficult for the adversary to breach the identity privacy and link privacy.

Similarly for structural queries [20], because of randomization, the adversary cannot simply exclude from those nodes that do not match the structural properties of the target. Instead, the adversary needs to consider the set of all possible graphs implied by $\widetilde{G}$ and k. Informally, this set contains any graph G_p that could result in $\widetilde{G}$ under k perturbations from G_p, and the size of the set is $\binom{m}{k}\binom{\binom{n}{2}-m}{k}$. The candidate set of a target node includes every node y if it is a candidate in some possible graph. The probability associated with a candidate y is the probability

of choosing a possible graph in which y is a candidate. The computation is equivalent to compute a query answer over a probabilistic database and is likely to be intractable.

We would emphasize that it is very challenging to formally quantify identity disclosure in the presence of complex background knowledge of adversaries (such as embedded subgraphs or graph metrics). Ying et al. [44] quantified the risk of identity disclosure (and link disclosure) when adversaries adopt one specific type of background knowledge (i.e., knowing the degree of target individuals). The node identification problem is that given the true degree d_α of a target individual α, the adversary aims to discover which node in the randomized graph $\widetilde{G}$ corresponds to individual α. However, it is unclear whether the quantification of disclosure risk can be derived for complex background knowledge based attacks.

4.2 Link Disclosure Analysis

Note that link disclosure can occur even if each vertex is K-anonymous. For example, in a K-degree anonymous graph, nodes with the same degree can form an equivalent class (EC). For two target individuals α and β, if every node in the EC of individual α has an edge with every node in the EC of β, the adversary can infer with probability 100% that an edge exists between the two target individuals, even if the adversary may not be able to identify the two individuals within their respective ECs. In [48], L. Zhang and W. Zhang described an attacking method in which the adversary estimates the probability of existing link (i, j) through the link density between the two equivalence classes. The authors then proposed a greedy algorithm aiming to reduce the probabilities of link disclosure to a tolerance threshold τ via a minimum series of edge deletions or switches.

In [45–47], the authors investigated link disclosure of edge-randomized graphs. They focused on networks where node identities (and even entity attributes) are not confidential but sensitive links between target individuals are confidential. The problem can be regarded as, compared to not releasing the graph, to what extent releasing a randomized graph $\widetilde{G}$ jeopardizes the link privacy. They assumed that adversaries are capable of calculating posterior probabilities.

In [45], Ying and Wu investigated the link privacy under randomization strategies (*Rand Add/Del* and *Rand Switch*). The adversary's prior belief about the existence of edge (i, j) (without exploiting the released graph) can be calculated as $P(a_{ij} = 1) = \frac{2m}{n(n-1)}$, where n is the number of nodes and m is the number of edges. For *Rand Add/Del*, with the

released graph and perturbation parameter k, the posterior belief when observing $\tilde{a}_{ij} = 1$ is $P(a_{ij} = 1|\tilde{a}_{ij} = 1) = \frac{m-k}{m}$.

An attacking model, which exploits the relationship between the probability of existence of a link and the similarity measure values of node pairs in the released randomized graph, was presented in [47]. Proximity measures have been shown to be effective in the classic link prediction problem [28] (i.e., predicting the future existence of links among nodes given a snapshot of a current graph). The authors investigated four proximity measures (common neighbors, Katz measure, Adamic/Adar measure, and commute time) and quantified how much the posterior belief on the existence of a link can be enhanced by exploiting those similarity values derived from the released graph which is randomized by the *Rand Add/Del* strategy. The enhanced posterior belief is given by

$$P(a_{ij} = 1|\tilde{a}_{ij} = 1, \tilde{m}_{ij} = x) = \frac{(1 - p_1)\rho_x}{(1 - p_1)\rho_x + p_2(1 - \rho_x)}$$

where $p_1 = \frac{k}{m}$ denotes the probability of deleting a true edge, $p_2 = \frac{k}{\binom{n}{2}-m}$ denotes the probability of adding a false edge, $\tilde{m}_{ij}$ denotes the similarity measure between node i and j in $\widetilde{G}$, and $\rho_x = P(a_{ij} = 1|\tilde{m}_{ij} = x)$ denotes the proportion of true edges in the node pairs with $\tilde{m}_{ij} = x$. The maximum likelihood estimator (MLE) of ρ_x can be calculated from the randomized graph.

The authors further theoretically studied the relationship among the prior beliefs, posterior beliefs without exploiting similarity measures, and the enhanced posterior beliefs with exploiting similarity measures. One result is that, for those observed links with high similarity values, the enhanced posterior belief $P(a_{ij} = 1|\tilde{a}_{ij} = 1, \tilde{m}_{ij} = x)$ is significantly greater than $P(a_{ij} = 1|\tilde{a}_{ij} = 1)$ (the posterior belief without exploiting similarity measures). Another result is that the sum of the enhanced posterior belief (with exploiting similarity measures) approaches to m, i.e.,

$$\sum_{i<j} P(a_{ij} = 1|\tilde{a}_{ij}, \tilde{m}_{ij}) \to m \quad \text{as} \quad n \to \infty,$$

while the sum of the prior beliefs and the sum of posterior beliefs (without exploiting similarity measures) over all node pairs equal to m. Notice that it is more desirable to quantify the probability of existing true link (i, j) via comprehensive information of $\widetilde{G}$, i.e., $P(a_{ij} = 1|\widetilde{G})$. However, this is very challenging.

A different attacking model was presented in [46]. It is based on the distribution of the probability of existence of a link across all possible graphs in the graph space $\mathcal{G}$ implied by G and k. If many graphs in $\mathcal{G}$

have an edge (i, j), the original graph is also very likely to have the edge (i, j). Hence the proportion of graphs with edge (i, j) can be used to denote the posterior probability of existence of edge (i, j) in the original graph. More details will be provided in Section 4.4.2.

4.3 Reconstruction

Recall that the edge randomization process can be written in the matrix form $\widetilde{A} = A+E$, where A ($\widetilde{A}$) is the adjacency matrix of the original (randomized) graph and E is the perturbation matrix. In the setting of randomizing numerical data, a data set U with m records of n attributes is perturbed to $\widetilde{U}$ by an additive noise data set V with the same dimensions as U. In other words, $\widetilde{U} = U + V$. Distributions of U can be approximately reconstructed from the perturbed data $\widetilde{U}$ using distribution reconstruction approaches (e.g., [2,3]) when some a-priori knowledge (e.g., distribution, statistics etc.) about the noise V is available. Specifically, Agrawal and Aggawal [2] provided an expectation-maximization (EM) algorithm for reconstructing the distribution of the original data from perturbed observations. However, it is unclear whether similar distribution reconstruction methods can be derived for network data. This is because 1) it is hard to define distribution for network data; and 2) the randomization mechanism for network data is based on the positions of randomly chosen edges rather than the independent random additive values for all entries for numerical data.

In [41], Wu et al. investigated the use of low rank approximation methods to reconstruct structural features from the graph randomized via *Rand Add/Del*. Let λ_i ($\tilde{\lambda}_i$) be A's ($\widetilde{A}$'s) i-th largest eigenvalue in magnitude whose eigenvector is $\boldsymbol{x}_i$ ($\tilde{\boldsymbol{x}}_i$). Then, the rank l approximations of A and $\widetilde{A}$ are respectively given by:

$$A_l = \sum_{i=1}^{l} \lambda_i \boldsymbol{x}_i \boldsymbol{x}_i^T \quad \text{and} \quad \widetilde{A}_l = \sum_{i=1}^{l} \tilde{\lambda}_i \tilde{\boldsymbol{x}}_i \tilde{\boldsymbol{x}}_i^T.$$

By choosing a proper l, Wu et al. [41] showed that $\widetilde{A}_l$ can preserve the major information of the original graph and filter out noises added in the rest dimensions. This is because real-world data is usually highly correlated in a low dimensional space while the randomly added noise is distributed (approximately) equally over all dimensions. In $\widetilde{A}_l$, those entries close to 1 are more likely to have true edges while those entries close to 0 are less likely to have edges. They simply derived the reconstructed graph $\hat{A}$ by setting the $2m$ largest off-diagonal entries in $\widetilde{A}_l$ as 1, and 0 otherwise. Empirical evaluations showed that more accurate

features can be reconstructed via the low rank approximation even when the magnitude of additive noise k equals to $0.8m$.

Note that the low rank approximation has been well investigated as a point-wise reconstruction method in the numerical setting. A spectral filtering based reconstruction method was first proposed in [22] to reconstruct original data values from the perturbed data. Similar methods (e.g., PCA based reconstruction method [21], SVD based reconstruction method [17]) were also investigated. All methods exploited spectral properties of the correlated data to remove the noise from the perturbed one. Preliminary results [41] showed that the accuracy of the reconstructed individual data (i.e., edge entries of the adjacency matrix) using the low rank approximation is not as good as that of the reconstructed numerical data.

We would emphasize that reconstruction methods on purely randomized graphs need further investigations so that more accurate analysis can be conducted on reconstructed graphs while individual privacy can be preserved. It is our conjecture that it is very hard, if not impossible, to figure out reconstruction methods on the released data randomized using K-anonymity schemes. This is because in K-anonymity based modification schemes, modified edge entries are not randomly chosen. For example, the K-degree scheme examines the degree sequence of nodes and chooses a subset of nodes (that violates the K-degree anonymity property) for edge modification.

4.4 Feature Preserving Randomization

Edge randomization may significantly affect the utility of the released randomized graph. To preserve utility, certain aggregate characteristics (a.k.a., feature) of the original graph should remain basically unchanged or at least some properties can be reconstructed from the randomized graph. However, as shown in [45], many topological features are lost due to randomization. In this section, we summarize randomization strategies that can preserve structural properties. We would emphasize that it is very challenging to quantify disclosures since the process of feature preserving strategies or generalization strategies is more complicated than that of randomization strategies.

Instead of randomizing the original graph via *Add/Del* or *Switch*, researchers also considered the problem of directly generating synthetic graphs given a set of features. We refer interested readers to a recent survey [10] and the references wherein for more details.

4.4.1 Spectrum Preserving Randomization. In [45], Ying and Wu presented a randomization strategy that can preserve the spec-

tral properties of the graph. The spectra of graph matrices have close relations with many important topological properties such as diameter, presence of cohesive clusters, long paths and bottlenecks, and randomness of the graph [36]. The authors aimed to preserve the data utility by preserving two important eigenvalues during the randomization: the largest eigenvalue of the adjacency matrix and the second smallest eigenvalue of the Laplacian matrix.

The authors showed that pure randomization tends to move the eigenvalues toward one direction, and the randomized eigenvalues can be significantly different from the original values. The two proposed algorithms, *Spctr Add/Del* and *Spctr Switch*, selectively pick up those edges that can increase (or decrease) the target eigenvalue by examining the eigenvector values of the nodes involved in the randomization, and apply the randomizing operation, which guarantees the randomized eigenvalues do not move far from the original value. Their empirical evaluations showed that the proposed algorithms can keep the spectral features as well as many topological features close to the original ones even when the magnitude of randomization is large.

Although they empirically showed that the spectrum preserving approach can achieve similar privacy protection as the random perturbation approach, however, they did not derive the formula of the protection measure for either *Spctr Add/Del* or *Spctr Switch* since the number of false edges in the randomization cannot be explicitly expressed.

4.4.2 Markov Chain based Feature Preserving Randomization. The degree sequence and topological features are of great importance to the graph structure. One natural idea is that it can better preserve the data utility if the released graph $\widetilde{G}$ preserves the original degree sequence and a certain topological feature, such as transitivity or average shortest distance. In [18, 46], the authors investigated switch based randomization algorithms that can preserve various properties of a real social network in addition to a given degree sequence.

To preserve data utility, data owners may want to preserve some particular feature $\boldsymbol{S}$ within a precise range in the released graph. All the graphs that satisfy the degree sequence $\boldsymbol{d}$ and the feature constraint $\boldsymbol{S}$ form a graph space $\mathcal{G}_{\boldsymbol{d},\boldsymbol{S}}$ (or $\mathcal{G}_{\boldsymbol{d}}$ if no feature constraint). Starting with the original graph, series of switches form a Markov chain that can explore the graph space $\mathcal{G}_{\boldsymbol{d},\boldsymbol{S}}$. Ying and Wu [46] proposed an algorithm that can generate any graph in $\mathcal{G}_{\boldsymbol{d},\boldsymbol{S}}$ with equal probability, and Hanhijarvi et al. [18] proposed an algorithm that generates a graph whose feature is close to the original value with high probability.

One concern on the privacy is that the feature constraint may reduce the graph space and increase the risk of privacy disclosure. In [46], Ying and Wu also studied how adversaries exploit the released graph as well as feature constraints to breach link privacy. The adversary can calculate the posterior probability of existence of a certain link by exploiting the graph space $\mathcal{G}_{\boldsymbol{d},\boldsymbol{S}}$. If many graphs in the graph space have link (i, j), the original graph is also very likely to have link (i, j), and hence the adversary's posterior belief about link (i, j) is given by

$$P[G(i,j) = 1|\mathcal{G}_{\boldsymbol{d},\boldsymbol{S}}] = \frac{1}{|\mathcal{G}_{\boldsymbol{d},\boldsymbol{S}}|} \sum_{G_t \in \mathcal{G}_{\boldsymbol{d},\boldsymbol{S}}} G_t(i,j).$$

The attacking model works as follows: knowing the degree sequence $\boldsymbol{d}$ and the feature constraint $\boldsymbol{S}$, the adversary generates N samples $G_t \in \mathcal{G}_{\boldsymbol{d},\boldsymbol{S}}$ $(t = 1, 2, \ldots, N)$ via the Markov chain that starts with the released graph $\widetilde{G}$ and converges to the uniform stationary distribution over the graph space. Then, $P[G(i,j) = 1|\mathcal{G}_{\boldsymbol{d},\boldsymbol{S}}]$ can be simply estimated by $\frac{1}{N}\sum_{t=1}^{N} G_t(i,j)$. The adversary can take the node pairs with highest posterior beliefs as candidate links. This attacking model works because the convergence of the Markov chain does not depend on the initial point. Their evaluations showed that some feature constraints can significantly enhance the adversary's attacking accuracy and the extent to which a feature constraint jeopardizes link privacy varies for different graphs.

5. Privacy Preservation via Generalization

To preserve privacy, both K-anonymity and randomization approaches modify the graph structure by adding/deleting edges and then release the detailed graph. Different from the above two approaches, generalization approaches can be essentially regarded as grouping nodes and edges into partitions called *super-nodes* and *super-edges*. The idea of generalization has been well adopted in anonymizing tabular data. For social network data, the generalized graph, which contains the link structures among partitions as well as the aggregate description of each partition, can still be used to study macro-properties of the original graph.

In [19], Hay et al. applied structural generalization approaches that groups nodes into clusters, by which privacy details about individuals can be hidden properly. To ensure node anonymity, they proposed to use the size of a partition as a basic guarantee against re-identification attacks. Their method obtains a vertex K-anonymous super-graph by aggregating nodes into super-nodes and edges into super-edges, such that, each super-node represents at least K nodes and each super-edge represents all the edges between nodes in two super-nodes. Because

only the edge density is published for each partition, it is impossible for the adversary to distinguish between individuals in partition. Note that more than one partition may be consistent with a knowledge query about target individual x. Hence, the size of a partition is used to provide a conservative guarantee against re-identification and there exists an improved bound on the size of candidate sets.

To retain utility, the partitions should fit the original network as closely as possible given the anonymity condition. The proposed method estimates fitness via a maximum likelihood approach. The likelihood is defined as one over the size of possible worlds implied by the partition. For any generalization $\mathcal{G}$, the number of edges in the super-node X is denoted as $c(X, X)$, the number of edges between X and Y is denoted as $c(X, Y)$, the set of possible worlds that are consistent with $\mathcal{G}$ is denoted by $\mathcal{W}(\mathcal{G})$ whose size is given by:

$$|\mathcal{W}(\mathcal{G})| = \prod_{X \in \mathcal{V}} \binom{\frac{1}{2}|X|(|X|-1)}{c(X,X)} \prod_{X,Y \in \mathcal{V}} \binom{|X||Y|}{c(X,Y)}$$

The likelihood for a graph $g \in \mathcal{W}(\mathcal{G})$ is then $1/|\mathcal{W}(\mathcal{G})|$. The partitioning of nodes is chosen so that the generalized graph satisfies privacy constraints and maximizes the utility $(1/|\mathcal{W}(\mathcal{G})|)$.

Their algorithm searches the approximate optimal partitioning, using simulated annealing [35]. Starting with a single partition containing all nodes, the algorithm proposes a change of state by splitting a partition, merging two partitions, or moving a node to a different partition. The movement from one partition to next valid partition is always accepted if it increases the likelihood and accepted with some probability if it decreases the likelihood. Search terminates when it reaches a local maximum.

The authors evaluated the effectiveness of structural queries on real networks from various domains and random graphs. Their results showed that networks are diverse in their resistance to attacks: social and communication networks tend to be more resistant than some random graph models (Erdos-Renyi and power-law graphs) would suggest, and hubs cannot be used to re-identify many of their neighbors.

One problem of this generalization approach is that since the released network only contains a summary of structural information about the original network (e.g., degree distribution, path lengths, and transitivity), users have to generate some random sample instances of the released network. As a result, uncertainty may arise in the later analysis since the samples come from a large number of possible worlds.

6. Anonymizing Rich Graphs

Real social network sources usually contain much richer information in addition to the simple graph structure. For example, in an online social network, the main entities in the data are individuals whose profiles can list lots of demographic information, such as age, gender and location, as well as other sensitive personal data, such as political and religious preferences, relationship status, etc. Between users, there are many different kinds of interactions such as friendship and email communication. Interactions can also involve more than two participants, e.g., many users can play a game together. Bhagat et al. [8] referred to the connections formed in the social networks as *rich interaction graphs*. Various queries on the network data are not simply about properties of the entities in the data, or simply about the pattern of the link structure in the graph, but rather on their combination. Thus it is important for the anonymization to mask the associations between entities and their interactions.

Notice that for rich social networks, a K-anonymous social network may still leak privacy. For example, if all nodes in a K-anonymous group are associated with some sensitive information, the adversary can derive that sensitive attribute of target individuals. Mechanism analogous to l-diversity [33] can be applied here. Several rich graph data models, which may contain labeled vertices/edges in addition to the structural information associated with the network, have been investigated in the privacy-preserving network analysis.

6.1 Link Protection in Rich Graphs

In [49], Zheleva et al. considered a graph model, in which there are multiple types of edges but only one type of nodes. Edges are classified as either sensitive or non-sensitive. The problem of link re-identification is defined as inferring sensitive relationships from non-sensitive ones. The goal is to attain privacy preservation of the sensitive relationships, while still producing useful anonymized graph data. They proposed to use the number of removed non-sensitive edges to measure the utility loss. Several graph anonymization strategies were proposed, including the removal of all sensitive edges and/or some non-sensitive edges, and the cluster-edge anonymization. In the cluster-edge anonymization approach, all the anonymized nodes in an equivalence class are collapsed into a single super-node and a decision is made on which edges to be included the collapsed graph. One feasible way is to separately publish the number of edges of each type between two equivalence classes.

The difference between the cluster-edge anonymization approach and the generalization approach in [19] is that the former aggregates edges by type to protect link privacy while the latter clusters vertices to protect node identities.

In [9], Campan and Truta considered an undirected graph model, in which edges are not labeled but vertices are associated with some attributes including identifier, quasi-identifier, and sensitive attributes. Those identifier attributes such as name and SSN are removed while the quasi-identifier and the sensitive attributes as well as the graph structure are released. To protect privacy in network data, they adopted the K-anonymity model for both the quasi-identifier attributes and the quasi-identifier relationship homogeneity. The goal is that any two nodes from any cluster are indistinguishable based on either their relationships or their attributes.

For structural anonymization, they proposed an edge generalization based method that does not insert or remove edges from the network data. They perform social network data clustering followed by anonymization through cluster collapsing. Specifically, the method first partitions vertices into clusters and attaches the structural description (i.e., the number of nodes and the number of edges) to each cluster. From the privacy standpoint, an original node within such a cluster is indistinguishable from other nodes. Then all vertices in the same cluster are made uniform with respect to the quasi-identifier attributes and the quasi-identifier relationship. This homogenization is achieved by using generalization, for both the quasi-identifier attributes and the quasi-identifier relationship. All vertices in the same cluster are collapsed into one single vertex (labeled by the number of vertices and edges in the cluster) and edges between two clusters are collapsed into a single edge (labeled with the number of edges between them). The method takes into account the information loss due to both the attribute generalization and the changes of structural properties. Users can tune the process to balance the tradeoff between preserving more structural information and preserving more vertex attribute information.

6.2 Anonymizing Bipartite Graph

Cormode et al. [11] studied a particular type of network data that can be modeled as bipartite graphs – there are two types of entities, and an association only exists between two entities of different types. One example is the pharmacy (customers buy products). The association between two nodes (e.g., who bought what products) is considered to be

private and needs to be protected while properties of some entities (e.g., product information or customer information) are public.

Their anonymization method can preserve the graph structure exactly by masking the mapping from entities to nodes rather than masking or altering the graph structure. As a result, analysis principally based on the graph structure is correct. Privacy is ensured in this approach because given a group of nodes, there is a secret mapping from these nodes to the corresponding group of entities. There is no information published that would allow an adversary to learn, within a group, which node corresponds to which entity.

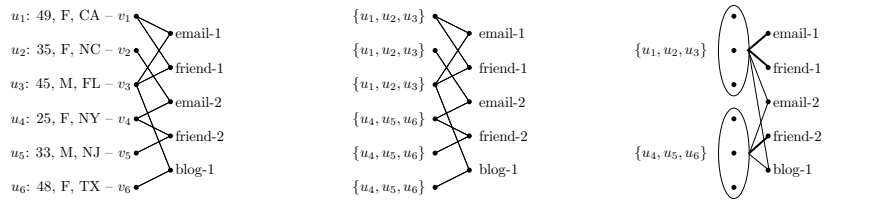
They evaluated the utility using three types of aggregate queries with increasing complexity for the bipartite graphs:

- Type 0 - Graph structure only: compute an aggregate over all neighbors of nodes in V that satisfy some P_n (i.e., predicates over solely graph properties of nodes), such as the average number of products bought by each customer.
- Type 1 - Attribute predicate on one side only: compute an aggregate for nodes in V satisfying P_a (i.e., predicates over attributes of the entities), such as the average number of products for NJ customers.
- Type 2 - Attribute predicate on both sides: compute an aggregate for nodes in V satisfying P_a and nodes in W satisfying P'_a, such as the total number of OTC products bought by NJ customers.

6.3 Anonymizing Rich Interaction Graph

In [8], Bhagat et al. adopted a flexible representation of rich interaction graphs which is capable of encoding multiple types of interactions between entities. Interactions involving large number of participants are represented by a hypergraph, denoted by $G(V, I, E)$. V is the node set. Each entity $v \in V$ has a hidden identifier u and a set of properties. Each entity in I is an interaction between/among a subset of entities in V. E is the set of hyperedges: for $v \in V$ and $i \in I$, an edge $(v, i) \in E$ represents node v participates in interaction i. One simple example of a hypergraph is shown in Figure 1.2(a).

The authors assumed that adversaries know part of the links and nodes in the graph. They presented two types of anonymization techniques based on the idea of grouping nodes in V into several classes. The authors pointed out that merely grouping nodes into several classes can not guarantee the privacy. For example, consider the case where the nodes within one class form a complete graph via a certain interaction.



(a) Original graph (b) Anonymized graph (c) Partitioning approach

Figure 1.2. The interaction graph example and its generalization results

Then, once the adversary knows the target is in the class, he can be sure that the target must participate in the interaction. The authors provided a safety condition, called *class safety* to ensure that the pattern of links between classes does not leak information: each node cannot have interactions with two (or more) nodes from the same group.

Their algorithm is briefly summarized as follows:

1 Sort the nodes according to attribute values.

2 Group the nodes in V into groups $\{C_i\}$ that satisfy the *class safety* property and $|C_i| \geq s$.

3 For node $v \in C_j$, replace the true identifier of v by a *label list* $l(v)$ containing $t \leq s$ identifiers, $l(v) = \{u_1, u_2, \ldots, u_t\}$. $l(v)$ contains the true identifier of v, and $\forall u_i \in l(v) \Rightarrow u_i \in C_j$.

After modification, graph G and the *label lists* are released. Figure 1.2(b) shows a special case where $s = t$ for the *label list*. In Figure 1.2(b), node v_1 has interactions with v_3 through an email and the friendship. This is allowed in the *class safety* property, as it allows two nodes to share multiple interactions, but prohibits a node having multiple friends in the same class. The authors also showed that the *label lists* are structured to ensure that the true identity cannot be inferred. Hence, the above procedures can greatly reduce the probability that an adversary can learn about other nodes and interactions through known nodes and interactions.

Note that the released graph contains the full topological structure of the original graph, some structural attacks such as the *active attack* and *passive attack* [4] can be applied here to de-anonymize the nodes in V. However, the adversary cannot further obtain the attributes of the target, for the attributes of those nodes within the same class are mixed together, which is similar to the anatomy approach [42] for the tabular database.

To prevent identity disclosure, the authors further proposed a solution, called *partitioning approach*, which groups edges in the anonymized graph and only releases the number of interactions between two groups, as illustrated in Figure 1.2(c). This method describes the number of interactions at the level of classes rather than nodes. The authors proved that this procedure guarantees that the adversary can correctly guess which nodes participate in the unknown links with probability at most $\frac{1}{s}$.

In term of the utility, the authors focused on the accuracy of aggregate queries on the graph data. They observed that if the nodes within one class have the same attribute values, the results of some queries can still be accurate, for the nodes of the class are either all included or all excluded in the result. Based on this idea, the proposed algorithms first sort all the nodes according to their attribute values, and then partition the nodes into classes that satisfy the *class safety* property. After partition, nodes within one class may not have exactly the same attribute values due to the *class safety* restriction, but they still have similar attribute values. The authors empirically showed that when the sorting order is appropriate, the query results based on the modified graph are not much different from the results based on the original graph.

6.4 Anonymizing Edge-Weighted Graph

Beyond the ongoing privacy-preserving social network analysis which mainly focuses on un-weighted social networks, in [13, 32], the authors studied the situations in which the network edges as well as the corresponding weights are considered to be private.

In [13], Das et al. considered the problem of anonymizing the weights of edges in the social network. The authors proposed a framework to re-assign weights to edges so that a certain *linear property* of the original graph can be preserved in the anonymized graph. A *linear property* is the property that can be expressed by a specific set of linear inequalities of edge weights. If the newly assigned edge weights also satisfy the set of linear inequalities, the corresponding *linear property* is also preserved. Then, finding new weight for each edge is a linear programming problem. The authors discussed two linear properties in details, single source shortest paths and all pairs shortest paths, and proposed the algorithms that can efficiently construct the corresponding linear inequality sets. Their empirical evaluations showed that the proposed algorithms can considerably improve the edge k-anonymity of the modified graph, which prevents the adversary to identify an edge by its weight.

In [32], Liu et al. also proposed two randomization strategies aiming to preserve the shortest paths in the weighted social network. The first one, which is easier to implement, is the Gaussian randomization multiplication strategy. The algorithm multiplies the original weight of each edge by an i.i.d. Gaussian random variable with mean 1 and variance σ^2. In the original graph, if the total weight of the shortest path between two nodes is much smaller than that of the second shortest path, the strategy can preserve the original shortest path with high probability. The authors further proposed the second strategy which can preserve a set of the target shortest paths or even all the shortest paths in the graph. The authors pointed out that all edges can be divided into three categories: the *all-visited edge* which belongs to all shortest paths, the *non-visited edge* which belongs to no shortest path, and the *partially-visited edge* which belongs to some but not all shortest paths. In order to preserve the target shortest paths, one can then reduce the weight of *all-visited edges*, increase the weight of *non-visited edges*, and perturb the weight of *partially-visited edges* within a certain range. The weight sum of a target shortest path is changed and is probably not the same as the original one, but the difference is minimized by the proposed greedy perturbation algorithm.

In both works of [13] and [32], the authors did not apply addition, deletion or generalization process to links or nodes. They only adjusted the weights of each links. However, their algorithms can be incorporated with some other graph modification algorithms.

7. Other Privacy Issues in Online Social Networks

We have restricted our discussion to the problem of privacy-preserving social network publishing so far. In this section, we give an overview about recent studies on other privacy issues in the real online social networks such as Facebook and MySpace.

7.1 Deriving Link Structure of the Entire Network

In [26], Korolova et al. considered a particular threat in which an adversary subverts user accounts to gain information about local neighborhoods in the network and pieces them together to build a global information about the social graph. It considered the case where no underlying graph is released, and, in fact, the owner of the network would like to keep the entire structure of the graph hidden from any one. The goal of the adversary is, rather than to de-anonymize partic-

ular individuals from that graph, to compromise the link privacy of as many individuals as possible. Specifically, the adversary determines the link structure of the graph based on the local neighborhood views of the graph from the perspective of several non-anonymous users.

Analysis showed that the number of users that need to be compromised in order to cover a constant fraction of the entire network drops exponentially with increase in the lookahead parameter l provided by the network data owner. Here a network has a lookahead l if a registered user can see all the links and nodes incident to him within distance l from him. For example, $l = 0$ if a user can see exactly who he links to; $l = 1$ if a user can see exactly the friends that he links to as well as the friends that his friends link to.

Each time the adversary gains access to a user account, he immediately covers all nodes that are at distance no more than the lookahead distance l enabled by the social network. In other words, he learns about all the edges incident to these nodes. Thus by gaining access to the account of user u, an adversary immediately covers all nodes that are within distance l of u. Additionally, he learns about the existence of all nodes within distance $l + 1$ from u. The authors studied several attacking strategies shown as below.

- Benchmark-Greedy: Among all users in the social network, pick the next user to bribe as the one whose perspective on the network gives the largest possible amount of new information. Formally, at each step the adversary picks the node covering the maximum number of nodes not yet covered.
- Heuristically Greedy: Pick the next user to bribe as the one who can offer the largest possible amount of new information, according to some heuristic measure. For example, Degree-Greedy picks the next user to bribe as the one with the maximum unseen degree, i.e., its degree minus the number of edges incident to it already known by the adversary.
- Highest-Degree: Bribe users in the descending order of their degrees.
- Random: Pick the users to bribe at random.
- Crawler: Similar to the Heuristically Greedy strategy, but choose the next node to bribe only from the nodes already seen (within distance $l+1$ of some bribed node). One example is Degree-Greedy-Crawler that picks, from all users already seen, the next user to bribe as the one with the maximum unseen degree.

Experiments on a 572, 949-node friendship graph extracted from LiveJoural.com indicated that 1) Highest-Degree yields the best performance while Random performs the worst; 2) in order to obtain 80% coverage of the graph using lookahead 2, Highest-Degree needs to bribe 6, 308 users while it only needs to bribe 36 users to obtain the same coverage using lookahead 3. The authors suggested that as a general rule, the social network owners should refrain from permitting a lookahead higher than 2. Data owner may also want to decrease their vulnerability of the social network by not showing the exact number of connections that each user has, or by varying the lookahead available to users based on their trustworthiness.

7.2 Deriving Personal Identifying Information from Social Networking Sites

Online network users often publish their profiles as well as their connections that contain vast amounts of personal and sometimes sensitive information (e.g., photo, birth date, phone number, current residence, various interests, and their friends). Acquisti and Gross in [16] studied the privacy risk associated with these networks. The user's profile information can be used to estimate a person's social security number and exposes his/her to identity theft. Their studies showed that only a small number of Facebook members change the default privacy preferences. As a result, users expose themselves to various physical and cyber risks, and make it extremely easy for third parties to create digital dossiers of their behavior. Their study quantified patterns of information revelation and inferred usage of privacy settings from actual field data.

8. Conclusion and Future Work

We surveyed recent studies on anonymization techniques for privacy-preserving publishing of social network data. The research and development of privacy-preserving social network analysis is still in its early stage compared with much better studied privacy-preserving data analysis for tabular data. We revisited the naive anonymization approach and several structural attacks which can be exploited on the naive anonymized graphs. We categorized the state-of-the-art anonymization methods on simple graphs in three main categories: K-anonymity based privacy preservation via edge modification, probabilistic privacy preservation via edge randomization, and privacy preservation via generalization. We then review anonymization methods on rich graphs. Since social network data is more complicated than tabular data, privacy preservation in social networks is much more challenging than privacy

preservation in tabular data. While ideas and methods can be borrowed from the well studied privacy preservation in tabular data, many serious efforts are greatly needed due to new challenges (see Section 1.2 and 1.3) associated with the network data. We present a set of recommendations for future research in this emerging area.

- Develop privacy models for graphs and networks. Investigate how well different strategies protect privacy (identity, link privacy, and attribute privacy) when adversaries exploit various complex background knowledge in their attacks. How to model various background knowledge and quantify disclosures when complex attacks are used needs to be investigated.
- Since how to preserve utility in the released graph is an important issue in privacy-preserving social network analysis, measures and methodologies need to be developed to quantify utility and information loss. It is important to develop workload-aware metrics that adequately quantify levels of information loss of graph data. Furthermore, various anonymization strategies need to be evaluated in terms of the tradeoff between privacy and utility.
- Existing studies except [52] do not consider dynamic releases. Many applications of evolutionary networks and dynamic social network analysis require publishing data periodically to support dynamic analysis. The "one-time" released network data from existing anonymization methods cannot guarantee privacy when adversaries collect historical information from multiple releases.
- Distributed privacy-preserving social network analysis based on secure multi-party computation [43]. Distributed privacy-preserving data analysis on tabular data has been well studied (e.g., [29]; refer to the book [1] for surveys). However, distributed privacy-preserving social network analysis has not been well reported in literature.
- Create a benchmark graph data repository. Researchers can compare and learn how different approaches work in terms of the privacy-utility tradeoff. The scalability issue needs to be studied and empirical evaluations need to be conducted on large social networks.

Acknowledgments

Authors Wu and Ying were supported in part by U.S. National Science Foundation IIS-0546027 and CNS-0831204.

References

[1] C. C. Aggarwal and P. S. Yu. *Privacy-Preserving Data Mining: Models and Algorithms*. Springer, 2008.

[2] D. Agrawal and C. Agrawal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the 20th Symposium on Principles of Database Systems*, 2001.

[3] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 439–450. Dallas, Texas, May 2000.

[4] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 181–190, New York, NY, USA, 2007. ACM Press.

[5] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 44–54, New York, NY, USA, 2006. ACM.

[6] J. Baumes, M. K. Goldberg, M. Magdon-Ismail, and W. A. Wallace. Discovering hidden groups in communication networks. In *ISI*, pages 378–389, 2004.

[7] T. Y. Berger-Wolf and J. Saia. A framework for analysis of dynamic social networks. In *KDD*, pages 523–528, 2006.

[8] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. Srivastava. Class-based graph anaonymization for social network data. In *Proc. of 35th International Conference on Very Large Data Base*, 2009.

[9] A. Campan and T. M. Truta. A clustering approach for data and structural anonymity in social networks. In *PinKDD*, 2008.

[10] D. Chakrabarti, C. Faloutsos, and M. McGlohon. *Graph Mining: Laws and Generators*. Springer, 2010.

[11] G. Cormode, D. Srivastava, T. Yu, and Q. Zhang. Anonymizing bipartite graph data using safe groupings. In *Proc. of VLDB08*, pages 833–844, 2008.

[12] L. da F. Costa, F. A. Rodrigues, G. Travieso, and P. R. V. Boas. Characterization of complex networks: A survey of measurements. *Advances In Physics*, 56:167, 2007.

[13] S. Das, Ömer Egecioglu, and A. E. Abbadi. Anonymizing edge-weighted social network graphs. Technical report, UCSB CS, March 2009.

[14] A. Fast, D. Jensen, and B. N. Levine. Creating social networks to improve peer-to-peer networking. In *KDD*, pages 568–573, 2005.

[15] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA*, 99(12):7821–7826, June 2002.

[16] R. Gross and A. Acquisti. Information revelation and privacy in online social networks (the Facebook case). *Proceedings of the Workshop on Privacy in the Electronic Society*, 2005.

[17] S. Guo, X. Wu, and Y. Li. Determining error bounds for spectral filtering based reconstruction methods in privacy preserving data mining. *Knowl. Inf. Syst.*, 17(2):217–240, 2008.

[18] S. Hanhijarvi, G. C. Garriga, and K. Puolamaki. Randomization techniques for graphs. In *Proc. of the 9th SIAM Conference on Data Mining*, 2009.

[19] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis. Resisting structural re-identification in anonymized social networks. In *VLDB*, 2008.

[20] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. Anonymizing social networks. *University of Massachusetts Technical Report*, 07-19, 2007.

[21] Z. Huang, W. Du, and B. Chen. Deriving private information from randomized data. In *Proceedings of the ACM SIGMOD Conference on Management of Data*. Baltimore, MA, 2005.

[22] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *Proc. of the 3rd Int'l Conf. on Data Mining*, pages 99–106, 2003.

[23] D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.

[24] J. M. Kleinberg. Challenges in mining social network data: processes, privacy, and paradoxes. In *KDD*, pages 4–5, 2007.

[25] Y. Koren, S. C. North, and C. Volinsky. Measuring and extracting proximity in networks. In *KDD*, pages 245–255, 2006.

[26] A. Korolova, R. Motwani, S. Nabar, and Y. Xu. Link privacy in social networks. In *Proceedings of the 24th International Conference on Data Engineering*, Cancun, Mexico, 2008.

[27] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *KDD*, pages 611–617, 2006.

[28] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 556–559, New York, NY, USA, 2003. ACM.

[29] Y. Lindell and B. Pinkas. Privacy preserving data mining. In *Advances in Cryptology (CRYPTO'00)*, pages 36–53. Springer-Verlag, 2000.

[30] K. Liu, K. Das, T. Grandison, and H. Kargupta. Privacy-preserving data analysis on graphs and social networks, 2008.

[31] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *Proceedings of the ACM SIGMOD Conference*, Vancouver, Canada, 2008. ACM Press.

[32] L. Liu, J. Wang, J. Liu, and J. Zhang. Privacy preservation in social networks with sensitive edge weights. In *SDM*, pages 954–965, 2009.

[33] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: privacy beyond k-anonymity. In *Proceedings of the IEEE ICDE Conference*, 2006.

[34] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *IEEE Security & Privacy '09*, 2009.

[35] S. Russell and P. Norvig. *Artifical Intelligence: A Modern Approach*. Pearson Education, 2003.

[36] A. Seary and W. Richards. Spectral methods for analyzing and visualizing networks: an introduction. *National Research Council, Dynamic Social Network Modelling and Analysis: Workshop Summary and Papers*, pages 209–228, 2003.

[37] M. Shiga, I. Takigawa, and H. Mamitsuka. A spectral clustering approach to optimally combining numericalvectors with a modular network. In *KDD*, pages 647–656, 2007.

[38] E. Spertus, M. Sahami, and O. Buyukkokten. Evaluating similarity measures: a large-scale study in the orkut social network. In *KDD*, pages 678–684, 2005.

[39] C. Tantipathananandh, T. Y. Berger-Wolf, and D. Kempe. A framework for community identification in dynamic social networks. In *KDD*, pages 717–726, 2007.

[40] S. White and P. Smyth. Algorithms for estimating relative importance in networks. In *KDD*, pages 266–275, 2003.

[41] L. Wu, X. Ying, and X. Wu. Reconstruction of randomized graph via low rank approximation. Technical report, UNC-Charlotte, SIS, 2009.

[42] X. Xiao and Y. Tao. Anatomy: Simple and effective privacy preservation. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, pages 139–150, September 2006.

[43] A. C. Yao. How to generate and exchange secrets. In *SFCS '86: Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, pages 162–167. IEEE Computer Society, 1986.

[44] X. Ying, K. Pan, X. Wu, and L. Guo. Comparisons of randomization and k-degree anonymization schemes for privacy preserving social network publishing. In *SNA-KDD '09: Proceedings of the 3rd SIGKDD Workshop on Social Network Mining and Analysis (SNA-KDD)*, 2009.

[45] X. Ying and X. Wu. Randomizing social networks: a spectrum preserving approach. In *Proc. of the 8th SIAM Conference on Data Mining*, April 2008.

[46] X. Ying and X. Wu. Graph generation with prescribed feature constraints. In *Proc. of the 9th SIAM Conference on Data Mining*, 2009.

[47] X. Ying and X. Wu. On link privacy in randomizing social networks. In *PAKDD*, 2009.

[48] L. Zhang and W. Zhang. Edge anonymity in social graphs. In *Proceedings of the 2009 International Conference on Social Computing*, 2009.

[49] E. Zheleva and L. Getoor. Preserving the privacy of sensitive relationships in graph data. In *PinKDD*, pages 153–171, 2007.

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

[51] B. Zhou, J. Pei, and W.-S. Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *SIGKDD Explorations*, 10(2), 2009.

[52] L. Zou, L. Chen, and M. T. Özsu. K-automorphism: A general framework for privacy preserving network publication. In *Proc. of 35th International Conference on Very Large Data Base*, 2009.