PRESERVING PRIVACY IN PUBLISHING SOCIAL NETWORK DATA

A DISSERTATION

Submitted in partial fulfillment of the
requirements for the award of the degree
of

INTEGRATED DUAL DEGREE
(Bachelor of Technology & Master of Technology)
in
COMPUTER SCIENCE AND ENGINEERING
(With Specialization in Information Technology)

By

BUDHWANI GULSHAN MAHESH

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)
JUNE, 2012

CANDIDATE'S DECLARATION

I hereby declare that the work being presented in the dissertation work entitled "**Preserving Privacy In Publishing Social Network Data**" towards the partial fulfillment of the requirement for the award of the degree of **Integrated Dual Degree in Computer Science and Engineering (with specialization in Information Technology)** and submitted to the **Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, India** is an authentic record of my own work carried out during the period from May, 2011 to June, 2012 under the guidance and provision of **Dr. Durga Toshniwal, Assistant Professor, Department of Electronics and Computer Engineering, IIT Roorkee.**

I have not submitted the matter embodied in this dissertation work for the award of any other degree and diploma.

Date: 12/06/2012

Place: Roorkee

(BUDHWANI GULSHAN MAHESH)

CERTIFICATE

This is to certify that the declaration made by the candidate above is correct to the best of my knowledge and belief.

Date: 12/6/12.

Place: Roorkee

Dr. Durga Toshniwal
E&CE Department
IIT Roorkee, India

ACKNOWLEDGEMENTS

I would like to take this opportunity to extend my heartfelt gratitude to my guide and mentor **Dr. Durga Toshniwal**, Assistant Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, for his trust in my work, his able guidance, regular source of encouragement and assistance throughout this dissertation work. I would state that the dissertation work would not have been in the present shape without his inspirational support and I consider myself fortunate to have done my dissertation under him.

I also extend my sincere thanks to **Dr. Padam Kumar**, Professor and Head of the Department of Electronics and Computer Engineering for providing facilities for the work.

I would like to thank all my friends who supported and encouraged me to finish this work.

Finally, I would like to say that I am indebted to my parents for everything that they have given to me. I thank them for sacrifices they made so that I could grow up in a learning environment. They have always stood by me in everything I have done, providing constant support, encouragement, and love.

BUDHWANI GULSHAN MAHESH

ABSTRACT

Recently, following the stupendous growth of various social networking sites, a vast amount of social network data has been collected. To put this data in use for commercial and research gains, more and more social network data have been published in one way or another. So an outright concern which comes up is preserving privacy in publishing social network data. With some local knowledge about individuals in a social network, an adversary may attack the privacy of some victims easily. Unfortunately, most of the previous studies on privacy preservation data publishing can deal with relational data only, and cannot be applied to social network data.

The basic objective of this dissertation was working towards preserving privacy in social network data. Specifically, two types of privacy attacks were identified: neighborhood attacks and friendship attacks. Bin Zhou and Jian Pei [10] proposed a scheme for anonymization of social networks against neighborhood attacks. Later, Tripathy and Panda[13] improved their algorithm for graph isomorphism by using adjacency matrix instead of min DFS code. Tai and Yang[14] provided a different solution for anonymization against friendship attacks. In this dissertation we propose a modified approach to their algorithms which anonymizes the social network against both neighborhood and friendship attacks simultaneously. Thus, the published social network will be privacy preserved against adversary attacks based on the vertex degree and neighborhood graph knowledge about individuals.

CONTENTS

LIST OF FIGURES AND TABLES

CHAPTER 1
INTRODUCTION

1.1 Social Network Publishing

A social network is an abstract concept, which describes the social relationships and interactions amongst individuals. In a concrete form, we can model a social network as a graph consisting of vertices and edges. Individuals/entities can be represented as vertices while their relationships and activities can be mapped to the edges. Recently, following the stupendous growth of various social networking sites, a vast amount of social network data has been collected. To put this data in use for commercial and research gains, more and more social network data have been published in one way or another.

1.1.1. Advantages

Often there exist various patterns in social relationships and interactions. Social network analysis is the field of study which extracts such patterns from the social network. Various application areas of social network analysis include detecting collusion and fraud, classifying the influence or popularity of individuals in organizational networks. Social network analysis can also be applied to study disease transmission in communities, the functioning of computer networks, and emergent behavior of physical and biological systems.

1.1.2. Privacy Concerns

Due to technological advances it is nowadays easier to gather the electronic records that describe social networks. However, agencies and researchers who collect such data often face with two undesirable problems. They can publish data for others to analyze, even though that analysis will create severe privacy threats [2], or they can withhold data because of privacy concerns, even though that makes further analysis impossible. Therefore, the goal should be to enable the useful analysis of social network data while protecting the privacy of individuals. Most of the recent works have focused on managing balance between privacy and utility in data publishing, but applicable only to some limited type of datasets. Some efforts in this direction are the k-anonymity [4] and its many variants, its extensions l-diversity [5], which are data perturbation techniques designed for tabular micro-data, which typically consists of a table of records, each of which describes an entity. These algorithms are not suitable to tackle the anonymization problem of social networks. A common assumption underlying all these techniques is that the records are independent and can be anonymized (more or less) independently. In contrast, social network data forms a graph of relationships between entities. Existing tabular perturbation techniques are not equipped to operate over graphs and they will tend to ignore and destroy important structural properties. Likewise, graph structure and background knowledge combine to threaten privacy in many new ways.

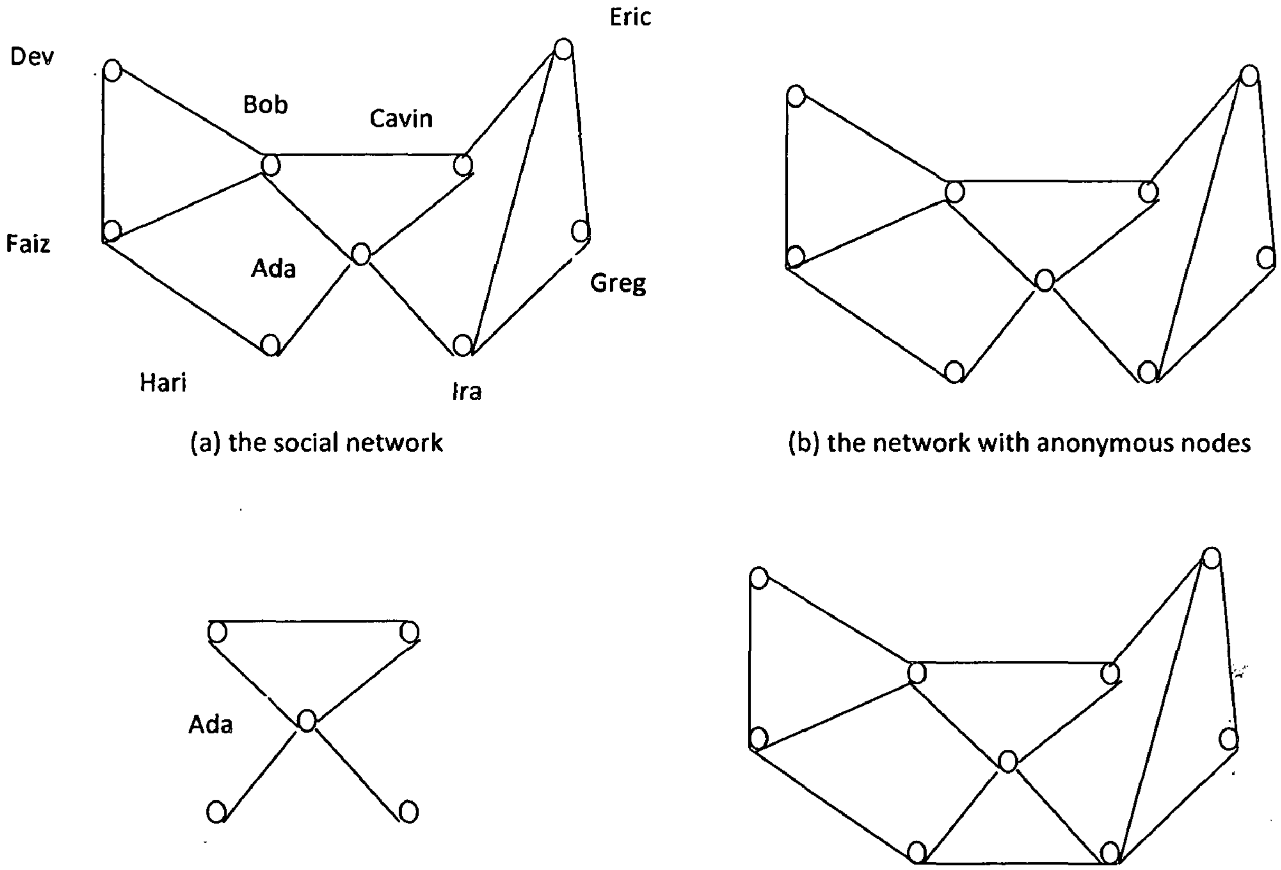
1.2 Motivation

1.2.1. Neighborhood Attacks:

With some local knowledge about individual vertices in asocial network, an adversary may attack the privacy of some victims. As a concrete example, consider a synthesized social network of "close-friends" shown in Figure 1.1(a). Each vertex in the network represents a person. An edge links two persons who are close friends.

Suppose the network is to be published. To preserve privacy, is it sufficient to remove all identities as shown in Figure 1.1(b)? Unfortunately, if an adversary has some knowledge about the neighbors of an individual, the privacy may still be leaked. If an adversary knows that Ada has two close friends who know each other, and has another two close friends who do not know each other, i.e., the 1-neighborhood graph of Ada as shown in Figure 1.1(c), then the vertex representing Ada can be identified uniquely in the network since no other vertices have the same 1-neighborhood graph. Similarly, Bob can be identified in Figure 1.1(b) if the adversary knows the1-neighborhood graph of Bob. [6]

Identifying individuals from released social networks intrudes privacy immediately. In this example, by identifying Ada and Bob, an adversary can even know from the released social network (Figure 1.1(b)) that Ada and Bob are close friends, and they share one common close friend. Other private information can be further derived such as how well a victim is connected to the rest of the network and the relative position of the victim to the center of the network.



(a) the social network

(b) the network with anonymous nodes

(c) the 1-neighborhood graph of Ada

(d) privacy preserved anonymous network

Fig 1.1: Illustration of neighborhood attacks

To protect the privacy satisfactorily, one way is to guarantee that any individual cannot be identified correctly in the anonymized social network with a probability higher than 1k ,where k is a user-specified parameter carrying the same spirit in the k-anonymity model [4]. By adding a noise edge linking Harry and Irene, the 1-neighborhood graph of every vertex in Figure 1(d) is not unique. An adversary with the knowledge of 1-neighborhood cannot identify any individual from this anonymous graph with a confidence higher than ½.[6].

1.2.2. Friendship Attacks:

A new type of attack, called a friendship attack, based on the vertex degree pair of an edge was identified by Tai and Yang [9]. Note that in a social networking website, such as Facebook, MySpace or Friendster, an adversary can acquire the number of friends of an individual. Moreover, the adversary can also extract the friendship relation between two individuals from the interaction information publicly available on the website. Therefore, using the vertex degrees of two individuals and their friendship relation, the adversary can issue a friendship attack on the published social network tore-identify the vertices corresponding to an individual and his friend as well as associated vertex information, such as hobbies, activities and religious beliefs.

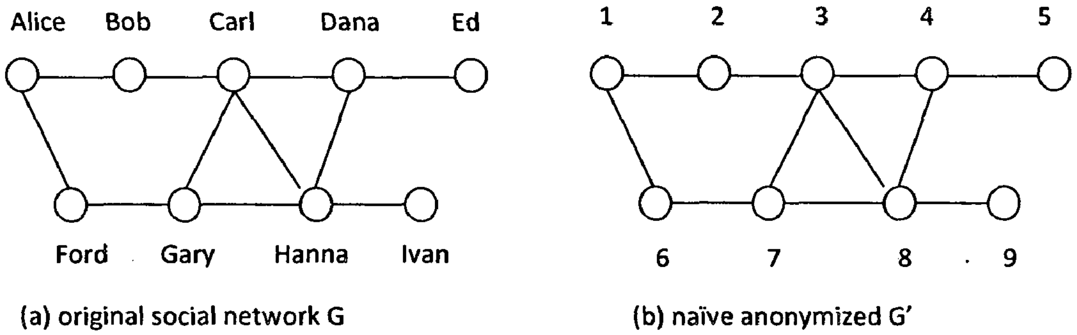


Fig 1.2: An example of the friendship attack

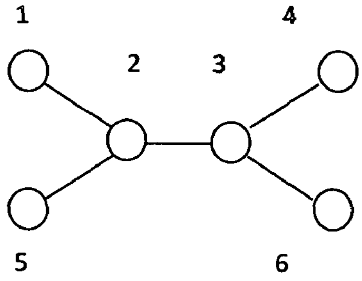


Fig 1.3: Example of anonymous graph (2^2degree)

Consider Figure 1.2 as an example. Suppose that a user's friend count is made publicly available on a social networking website. With the vertex degree information, an adversary cannot uniquely re-identify anyone from the naive anonymized social networking Figure 1.2(b).

However, if the adversary also knows that Bob and Carl are friends, both Bob and Carl are uniquely identified by the vertex degree pair (2, 4). This example illustrates that it is possible to launch an effective attack and identify individuals as long as the friendship information on the social networking websites can be obtained.[9]

1.3 Problem Statement

"The aim is to anonymize the published social network data for preserving privacy against neighborhood and friendship attacks, such that the network characteristics are retained and aggregate network queries can be executed accurately."

1.3.1 Problem Description

Problem Formulation

The aim is to protect the privacy of the individuals, represented as vertices in the social network graph. We need to formulate the problem with the above aim. It requires the representation of the social network, identifying the privacy information to be secured and specifying the background information.. Consider a graph G= (V, E) and its published anonymized graph G'= (V', E'). The vertices in the network should not be identified in the anonymized graph. For a positive integer k, the privacy of u is preserved in G if it cannot be re identified in G' with a probability larger than 1/k. Secondly, we define the background knowledge. In our case, the adversary will be having information about the neighborhood of a few vertices and also the degree of vertices. The purpose of the anonymized graph will be to answer aggregate network queries.

Problem Definition

Once we formulate the problem, we begin to solve it with certain criterion which defines the problem. Given a social network G, we want to compute an anonymization G' such that

(1) G' is k-anonymous to neighborhood and friendship attacks;

(2) each vertex in G is anonymized to a vertex in G' and G' does not contain any fake vertex;

(3) every edge in G is retained in G' and

(4) G' can be used to answer aggregate network queries as accurately as possible.

1.4 Organization of Dissertation

This report comprises of six chapters including this chapter that introduces the topic and states the problem. The rest of the dissertation report is organized as follows.

Chapter 2 provides a brief description of literature review on privacy preservation in social network publication. The other topics include definitions for social network anonymization, various privacy preserving schemes in social network data publication, existing system for social network anonymization against neighborhood attacks and research gaps.

Chapter 3 provides a detailed description of the proposed scheme for preserving privacy of individuals in the published social network against neighborhood and friendship attacks.

Chapter 4 gives the brief description of the implementation of the proposed scheme and results and performance evaluation.

Chapter5 concludes the work and gives the directions for future work.

CHAPTER 2
LITERATURE REVIEW

2.1. Social Networks

From the point of view of data mining, *a social network is a heterogeneous and multi-relational data set represented by a graph*. The graph is typically very large, with nodes corresponding to objects and edges corresponding to links representing relationships or interactions between objects. Both nodes and links have attributes. Objects may have class labels. Links can be one-directional and are not required to be binary.

2.2 Social Network Analysis

Social network analysis provides a rich and systematic means of assessing such network by mapping and analyzing relationships among people, teams, departments or even the entire organization. Organizations are considered as a network of individuals and researchers have used network analysis to map information flow as well as relational characteristics among strategically important groups to improve knowledge creation and sharing. Mapping and understanding social networks within an organization is a mean to understand how social relationships may affect business processes.[1]

2.2.1. Indices and Dimensions:

Various structural measures can be applied to social networks. These structures are characterized by relationships, entities, context, configurations, and temporal stability. Some of the indices and dimensions that express outcomes of network are:

Size Density and Degree.

Size is critical for the structure of social relations due to each actor has limited resources for building and maintaining ties. The degree of an actor is defined as the sum of the connections between the actor and others. The density measurement can be used to analyze the connectivity and the degree of nodes and links in a social network. [1]

Centrality:

The centrality of a social network is a measurement that is used to measure the betweenness and closeness of the social network. The measure of centrality which can be used to identify who have the most connections to others in the network (high degree) or whose departure would cause the network to fall apart.

Structural hole:

The structural hole is also a measurement of social network analysis, which can be used to discover the holes in a social network and by this to fill the hole and expand the social network.

Reachability:

The reachability can be used to analyze how to reach a node from another node in the social networks. An actor is reachable by another if there exists any set of connections by which we can trace from the source to the target actor, regardless of how many others fall between them.

Distance:

Because most individuals are not usually connected directly to most other individuals in a population, it can be quite important to go beyond simply examining the immediate connections of actors, and the overall density of direct connections in populations. Walk, trail and path are basic concepts to develop more powerful ways of describing various aspects of the distances among actors in a network. [3]

2.2.2. Characteristics of Social Networks:

Social networks are rarely static. Their graph representations evolve as nodes and edges are added or deleted over time. In general, social networks tend to exhibit the following phenomena:

Densification power law:

Previously, it was believed that as a network evolves, the number of degrees grows linearly in the number of nodes. This was known as the constant average degree assumption. However, extensive experiments have shown that, on the contrary, networks become increasingly dense over time with the average degree increasing (and hence, the number of edges growing super-linearly in the number of nodes). The densification follows the densification power law (or growth power law), which states

$$e(t) \propto n(t)^a \qquad [2.1]$$

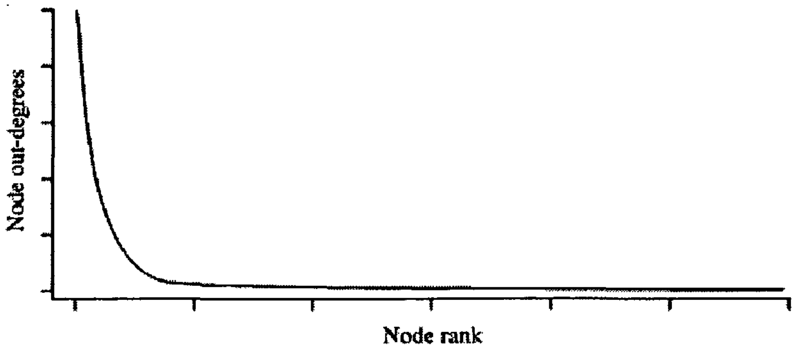
where e(t) and n(t), respectively, represent the number of edges and nodes of the graph at time t, and the exponent a generally lies strictly between 1 and 2. Note that if a=1, this corresponds to constant average degree over time, whereas a = 2 corresponds to an extremely dense graph where each node has edges to a constant fraction of all nodes.[3]

Shrinking diameter:

It has been experimentally shown that the effective diameter tends to decrease as the network grows. This contradicts an earlier belief that the diameter slowly increases as a function of network size. As an intuitive example, consider a citation network, where nodes are papers and a citation from one paper to another is indicated by a directed edge. The out-links of a node, v (representing the papers cited by v), are "frozen" at the moment it joins the graph. The decreasing distances between pairs of nodes consequently appears to be the result of subsequent papers acting as "bridges" by citing earlier papers from other areas.[3]

Heavy-tailed out-degree and in-degree distributions:

The number of out-degrees for a node tends to follow a heavy-tailed distribution by observing the power law, $1/n^a$, where n is the rank of the node in the order of decreasing out-degrees and typically, 0<a<2 . The smaller the value of a, the heavier the tail. This phenomena is represented in the preferential attachment model, where each new node attaches to an existing network by a constant number of out-links, following a "rich-get-richer" rule. The in-degrees also follow a heavy-tailed distribution, although it tends be more skewed than the out-degrees distribution.[3]



The number of out-degrees (y-axis) for a node tends to follow a heavy-tailed distribution. The node rank (x-axis) is defined as the order of deceasing out-degrees of the node.

Fig 2.1: Graph(Node out-degrees vs Node rank) [3]

A Forest Fire model for graph generation was proposed, which captures these characteristics of graph evolution over time. It is based on the notion that new nodes attach to the network by "burning" through existing edges in epidemic fashion. It uses two parameters, forward burning probability, p, and backward burning ratio, r, which are described below. Suppose a new node, v, arrives at time t. It attaches to G_t , the graph constructed so far, in the following steps:[3]

1. It chooses an ambassador node, w, at random, and forms a link to w.

2. It selects x links incident to w, where x is a random number that is binomially distributed with mean $(1- p)^{-1}$. It chooses from out-links and in-links of w but selects in-links with probability r times lower than out-links. Let w_1,w_2, ... ,w_x denote the nodes at the other end of the selected edges.

3. Our new node, v, forms out-links to w_1,w_2, ... ,w_x and then applies step 2 recursively to each of w_1,w_2, ... ,w_x. Nodes cannot be visited a second time so as to prevent the construction from cycling. The process continues until it dies out.

Several earlier models of network evolution were based on static graphs, identifying network characteristics from a single or small number of snapshots, with little emphasis on finding trends over time. The Forest Fire model combines the essence of several earlier models, while considering the evolution of networks over time. The heavy-tailed out-degrees property is observed in that, owing to the recursive nature of link formation, new nodes have a good chance of burning many edges and thus producing large out-degrees. The heavy-tailed in-degrees property is preserved in that Forest Fire follows the "rich-get-richer" rule: highly linked nodes can easily be reached by a new node, regard-less of which ambassador the new node starts from. The flavor of a model known as the copying model is also observed in that a new node copies many of the neighbors of its ambassador. The densification power law is upheld in that a new node will have many links near the community of its ambassador, a few links beyond this and much fewer farther away. Rigorous empirical studies found that the shrinking diameter property was upheld. Nodes with heavy-tailed out-degrees may serve as "bridges" that connect formerly disparate parts of the network, decreasing the network diameter.[3]

2.3. Privacy Preserving Schemes in Social Network Data Publication:

Privacy becomes a more and more serious concern in many applications. The development of techniques that incorporate privacy concerns has become a fruitful direction for database and data mining research. In this section, we overview the various existing privacy preserving schemes in publishing social networks.

2.3.1. Min-DFS code based Anonymization

The algorithm suggested by Zhou and Pei [10] solves the anonymization problem using a two step method as summarized below. At first the neighborhoods of vertices are extracted and grouped and then the vertices are anonymized.

2.3.1.1. Neighborhood Extraction and Vertex Organization

The neighborhood of each vertex is extracted and the different components are separated. As the requirement is to anonymize all graphs in the same group to a single graph, isomorphism tests are conducted. For this purpose, for every component of the vertex the following steps are performed. At first all possible DFS trees are constructed for the component and their DFS codes are obtained from which the minimum DFS code is selected. This code represents the component. Minimum DFS code has a nice property [10]: two graphs G and G' are isomorphic if and only if DFS (G) = DFS (G'). Then neighborhood component code order is used to obtain single code for each vertex.

2.3.1.2. Anonymization Algorithm

Anonymization is done by taking the vertices from the same group. If the match is not found, the cost factor is used to decide the pair of vertices to be considered.

Anonymization of Social Network Algorithm ([10])

```
initialize G' = G;
mark vi $\in$ V (G) as "unanonymized";
sort vi $\in$ V (G) as Vertex List in neighborhood size descending order;
WHILE (Vertex List $\varphi \neq$ ) DO
```

```
    let Seed Vertex = VertexList.head () and remove it from Vertex List;
    FOR each vi ∈ Vertex List DO
        calculate Cost (Seed Vertex vi) using the anonymization method for two vertices;
    END FOR
    IF (VertexList.size () ≥ 2k - 1) DO
        let Candidate Set contain the top k - 1 vertices with the smallest Cost;
    ELSE
        let Candidate Set contain the remaining unanonymized vertices;
    suppose Candidate Set= {u1...um} anonymize Neighbor(Seed Vertex) and Neighbor(u1)
    FOR j = 2 to m DO
        anonymize Neighbor(uj) and {Neighbor(SeedVertex), Neighbor(u1).......Neighbor(uj-1)} mark them as "anonymized
        update Vertex List;
    END FOR
END WHILE
```

2.3.1.3. Minimum DFS code notation

To solve the uniqueness problem, a minimum DFS code notation is proposed in [19]. For any connected graph G, let T be a DFS-tree of G. Then, an edge is always listed as (vi; vj) such that i < j. A linear order Á on the edges in G can be defined as follows. Given edges e = (vi; vj) and e0= (vi0; vj0). e < e0 if

(1) when both e and e0 are forward edges (i.e., in DFS-tree T), j < j0 or (i > i0 ^ j = j0);

(2) when both e and e0 are backward edges (edges not in DFS-tree T), i < i0 or (i = i0 ^ j < j0);

(3) when e is a forward edge and e0 is a backward edge, j · i0; or

(4) when e is a backward edge and e0 is a forward edge, i < j0.

2.3.1.4. Definitions and Preliminaries

The various terms that are to be used in the rest of the dissertation have been described here.

Definition1 Modeling a social network

A social network can be modeled as a simple graph, G= (V, E) where, V is the set of vertices of the graph, E is the edge set.

Definition2 Neighborhood of a vertex and neighborhood component

In a social network G, the neighborhood of u ∈ V (G) is the induced sub graph of the neighbors of u, denoted by NeighborG(u) = G(Nu) where Nu = {v|(u, v) ∈ E(G)}. The components of the neighborhood graph of a vertex are the neighborhood components.[13]

Definition3 d- Neighborhood

The d-Neighborhood graph of a vertex u includes all the vertices that are within the distance'd' from the vertex u.

Anonymization Preliminaries

The major challenge in designing anonymization techniques is that adding edges or changing the labels of the vertices may affect the neighborhoods of some other vertices as well as the properties of the networks. The k-anonymity requires that each vertex u ∈V (G) is grouped with at least (k -1) other vertices such that their anonymized neighborhoods are isomorphic.[10]

Loss due to addition of edges: This is measured by the total number of edges added and the number of vertices those are linked to the anonymized neighborhood for anonymization. As an example, consider two vertices u1, u2 ∈ V (G), where G is a social network. Suppose

Neighbor G (u1) and Neighbor G (u2) are generalized to Neighbor G' (A(u1)) and Neighbor G' (A(u2)), which are isomorphic. The anonymization cost is given by [10]

Cost (u, v) = γ*(No of new edges added into G')

+ δ*(No of vertices included into neighborhoods of 'u' and 'v' due to addition of

edges), where γ and δ are the weights proposed by the user.

2.3.2. Adjacency Matrix based Anonymization:

2.3.2.1. Similarity Check for Component

First, we separate neighborhood graphs of all the vertices into their components and represent them in the form of adjacency matrices. The adjacency matrix is constructed in the decreasing order of the vertices in the component. In case of two or more vertices having the same degree the ordering is done arbitrarily.[13]

When two components have the same degree and same adjacency matrices, they are isomorphic according to their structure. If these two components have same labels too, then they are isomorphic. Else we generalize their labels to their parent label. When the components have different number of vertices, then the similarity between them is done by comparing the first sub matrices of the adjacency matrices of the components with highest number of vertices. If the sub matrices do not match, we can add vertices and edges for anonymization or making them isomorphic. [13]

2.3.2.2. Anonymize two Vertices

Consider two neighborhoods of u, v ∈ V(G) as shown in Figure 2.2. The components in each neighborhood are ordered in descending order and are grouped and named based on the number of vertices of the component. Thus [C1(u)], [C2(u)] and [C3(u),C4(u)] are the three groups formed from Neighborhood(u) and [C1(v)], [C2(v)], [C3(v)] and [C4(v)] are the four groups formed from Neighborhood(v). [13]

The adjacency matrices for all the components are compared for similarity in the following order:.

1. Components from the two neighborhoods are first compared in the respective groups. In figure 2.2, C1(u) and C1(v) are the components of the respective 1-vertex groups and have same adjacency matrices. We get the corresponding anonymized neighborhood C1(a).

2. The next respective groups of 2-vertices in both the groups are considered for anonymization. C2 (u) and C2(v) are similar in all respects, so they are simply anonymized without any changes. The corresponding anonymized neighborhood is C2 (a).[13]

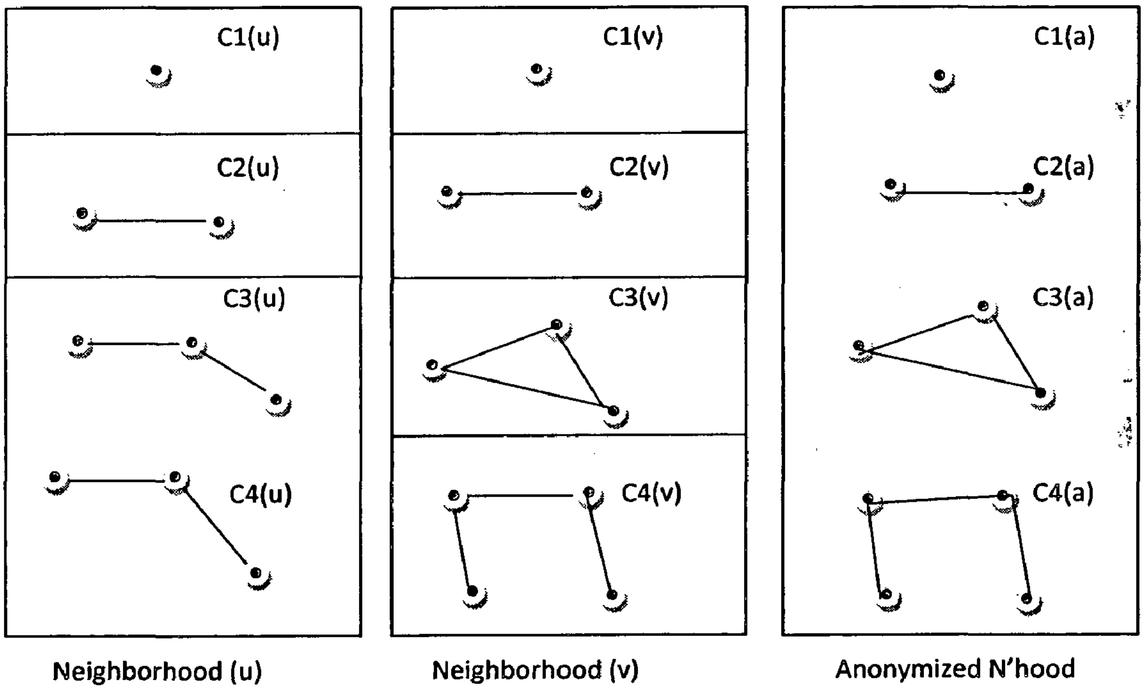


Figure 2.2 : Illustration of anonymization of two neighborhoods

3. The components C3(u) and C4(u) are compared with C3(v). Here, the adjacency matrices are not similar. So, one of the components (in this case C3(u) is considered randomly. Thus, C3(u) and C3(v) are considered for anonymization. Since there is an edge deficiency in C3(u), an extra edge is added to make it similar to C3(v). The resultant anonymized component is C3(a).

4. The final component 'C4 (u)' is compared with C4(v). Since C4(u) has one vertex deficient, a vertex w ∈ V(G), that is neither in Neighborhood(u) or Neighborhood(v) is brought into Neighborhood(u) and is added to C4(u) to make C4(u) and C4(v) similar. the resultant anonymized component is as C4(a). Thus the final anonymized neighborhood is as shown in Figure 2.2 .

C. Network Anonymization Procedure

1. Neighborhood Extraction: For all the vertices of the graph, G the vertices that fall in its d-neighborhood are considered. The neighborhood components are obtained for each of the vertex neighborhood. [13]

2. First, mark all vertices in the network as "unanonymized". Maintain a list Vertex List of "unanonymized" vertices according to the descending order of the number of vertices in the neighborhood. The vertices with the same number of vertices in neighborhood are arranged arbitrarily.

3. Iteratively, we pick the first vertex Seed Vertex in the list Vertex List. The anonymization cost of Seed Vertex and any other vertices in Vertex List is calculated using the anonymization method for two vertices. B. If the number of unanonymized vertices in Vertex List is at least 2k-1, we select a Candidate Set of the top k – 1 vertices in the Vertex List with the smallest anonymization cost.

4. It is not possible that every vertex in a graph can find at least one other vertex with isomorphic neighborhoods. So, a factor known as 'Anonymization Quality Measure' also known as 'Anonymization Cost' is calculated for every pair of vertices that do not find a match. The vertices with the minimum cost difference can be grouped together for anonymization.[13]

5. The Seed Vertex and the vertices in the Candidate Set = {u1, u2, ...,um} are anonymized in turn using the anonymization method for two vertices discussed in Section V. B. The anonymization of Seed Vertex and u1 is straightforward. After these two vertices are anonymized, their neighborhoods are identical. When we anonymize them with respect to u2, any change (e.g., adding an edge or a neighbor node) to the neighborhood of Seed Vertex will be

applied to u1 as well, so that the neighborhoods of Seed Vertex, u1 and u2 are same. The process continues until the neighborhoods of Seed Vertex and u1, u2, ... um are anonymized.[13]

6. During the anonymization of a group of vertices, some changes may occur to some other vertices v that have been marked as "anonymized" in another group (e.g., adding edges between an anonymized vertex and a vertex being anonymized based on vertex matching). In order to maintain the k-anonymity for these vertices, we apply the same changes to every other k-1 vertices having the isomorphic neighborhoods as v. Once those k vertices are changed, they are marked as "unanonymized" and inserted into the Vertex List again.

7. When the number of unanonymized vertices in Vertex List is less than 2k, to satisfy the k-anonymity, the remaining vertices in Vertex List have to be considered together for anonymization. They are added to the Candidate Set in a batch. The social network anonymization algorithm continues until all the vertices in the graph are marked as "anonymized".[13]

2.3.3. Others:

One of the privacy concerned problems is publishing microdata for public use, which has been extensively studied recently. A large category of privacy attacks is to re-identify individuals by joining the published table with some external tables modeling the background knowledge of users. To battle this type of attacks, the mechanism of k-anonymity was proposed in [2], [4]. Specifically, a data set is said to be k-anonymous (k , 1) if, on the quasi-identifier attributes (i.e., the minimal set of attributes in the table that can be joined with external information to re-identify individual records), each record is indistinguishable from at least k - 1 other records within the same data set. The larger the value of k, the better the privacy is protected.

Machanavajjhala et al. [5] showed that a k-anonymized dataset has some subtle but severe privacy problems due to the lack of diversity in the sensitive attributes. In particular, they showed that, the degree of privacy protection does not really depend on the size of the quasi-identifier attribute set. Instead, it is determined by the number of distinct sensitive values associated with each quasi-identifier attribute set. The observation leads to the notion of l-diversity [5].

Beyond microdata, some other data sources such as social network data also have privacy concerns when they are published for public use. Typically, social network data can be represented as a graph, in which vertices correspond to people or other social entities, and edges correspond to social links between them [1]. As a first step to hide information about social entities while preserving the global network properties, the released social network data has to go through the anonymization procedure which replaces social entity names with meaningless unique identifiers. Although this kind of anonymization can exactly preserve the unannotated structure of the social network, it may still leak a lot of information.

Attacks in social network data can be regarded as one kind of link mining[9]. Specifically, as a pioneer work about privacy in social network data, Backstrom et al.[9] described a family of attacks based on random graph theory. For example, an attacker may plant some well constructed sub-structures associated with the target entities in advance. Once the social network data is collected and published, the attacker can first try to identify the planted structures and thus peek the linkage between the target vertices. However, there is no practical solution proposed into counter those attacks.

The attacks proposed in [9] are different from the neighborhood attacks addressed in this dissertation. The attacks in [9] need to plant a set of deliberative structures before the social network data is anonymized, which is a task hard to achieve in some situations. As shown before, even without planting deliberative structures, the released social network data is still in danger, as neighborhood attacks are still possible.

Wang et al. [6] adopted description logic as the underlying knowledge representation formalism, and proposed some metrics of anonymity for assessing the risk of breaching confidentiality by disclosing social network data. However, they did not give any anonymization algorithms for social network data.

Hay et al. [7] presented a framework for assessing the privacy risk of sharing anonymized network data. They modeled the adversaries' background knowledge as vertex requirement structural queries and subgraph knowledge structural queries, and proposed a privacy requirement k-candidate anonymity which is similar to k-anonymity in tabular data. They developed a random graph perturbation method by randomly deleting or adding edges to anonymize a social network. Their model assumes that the nodes and the edges in a social network are not labeled.

Zheleva et al. [8] proposed a model different from ours. They focused on social networks where nodes are not labeled but edges are labeled. Some types of edges are sensitive and should be hidden. They provided the edge anonymization methods based on edge clustering and removal to prevent link re-identification.

Zhou and Pei proposed an anonymization technique for social networks to prevent the neighborhood attacks [10]. They have anonymized the social network using depth-first search (DFS for short) codes and minimum DFS codes. This approach gives a simpler solution but is limited to the case that the adversary has information about the immediate neighbor only.

More recently, Thomson and Yao [11] have presented two clustering algorithms for clustering undirected graphs that group similar graph nodes into clusters with a minimum size constraint. Also, they have developed an inter-cluster matching method for anonymizing social networks by strategically adding and removing edges based on the social role of the nodes.

2.4. Research Gaps

- Only 1-neighborhoods of the vertices were handled while using Min DFS code technique for node isomorphism. It is very interesting and could be desirable in some applications that d-neighborhoods (d > 1) are protected
- The anonymized social network was still prone to friendship attacks in which the adversary has the knowledge of the degree of all nodes.
- A k-anonymous social network still may leak privacy. If an adversary can identify a victim in a group of vertices anonymized in a group, but all are associated with some sensitive information, then the adversary still can know that sensitive attribute of the victim

CHAPTER 3
PROPOSED SCHEME FOR PRIVACY PRESERVATION IN SOCIAL NETWORKS

We have so far studied how the privacy of individuals can be breached in published social networks through neighborhood attacks (using knowledge about neighborhood graph) and friendship attacks (using knowledge about vertex degree). Further, we extensively considered the solution provided by Zou and Pei[10] to cope with neighborhood attacks by anonymizing the social network on the lines of K-anonymity model. But this anonymized network was still vulnerable to friendship attacks.

In this dissertation, we hereby propose a solution to anonymize the social network in such a manner that it is immune to both neighborhood and friendship attacks. We provide a solution on the lines of alternate algorithm used by Tripathi and Panda[13] to anonymize the social network wherein they used adjacency matrix to check for isomorphism and anonymization instead of minimum DFS code used by Zou and Pei[10].

Instead of anonymizing 1-neighborhood graph of the vertices, if we consider anonymizing 2-neighborhood graph of the vertices i.e the nodes connected to a vertex upto a distance of 2 nodes; we overcome the problem of friendship attacks along with neighborhood attacks. The algorithm used by Zou and Pei [10] had a shortcoming that it can only be used for 1-neighborhood of vertices. But if consider the alternate approach in social network anonymization by Tripathi and Panda[13] of using adjacency matrix based isomorphism we can anonymize the nodes upto the depth of 2 nodes(2-neighborhood). We hereby provide a modified approach to the their solution for the objective of anonymizing social network for simultaneously preventing neighborhood attacks and friendship attacks.

In our approach, we distinguish vertices into two lists α and β. All the nodes in the β list are directly connected to the nodes in the α list. We begin to k- anonymize vertices that belong to list α only. Instead of anonymizing direct neighborhood, we take into consideration neighborhoods upto the depth of 2 nodes. Once all the nodes in the α list are k-anonymized up to the 2-neighborhood; we get a social network graph in which :

1. all the nodes are k-anonymized upto 1-neighborhood thereby making the social network graph immune to neighborhood attacks.
2. all the nodes are immune to friendship attacks too.

3.1. Proposed Framework for Social Network Anonymization

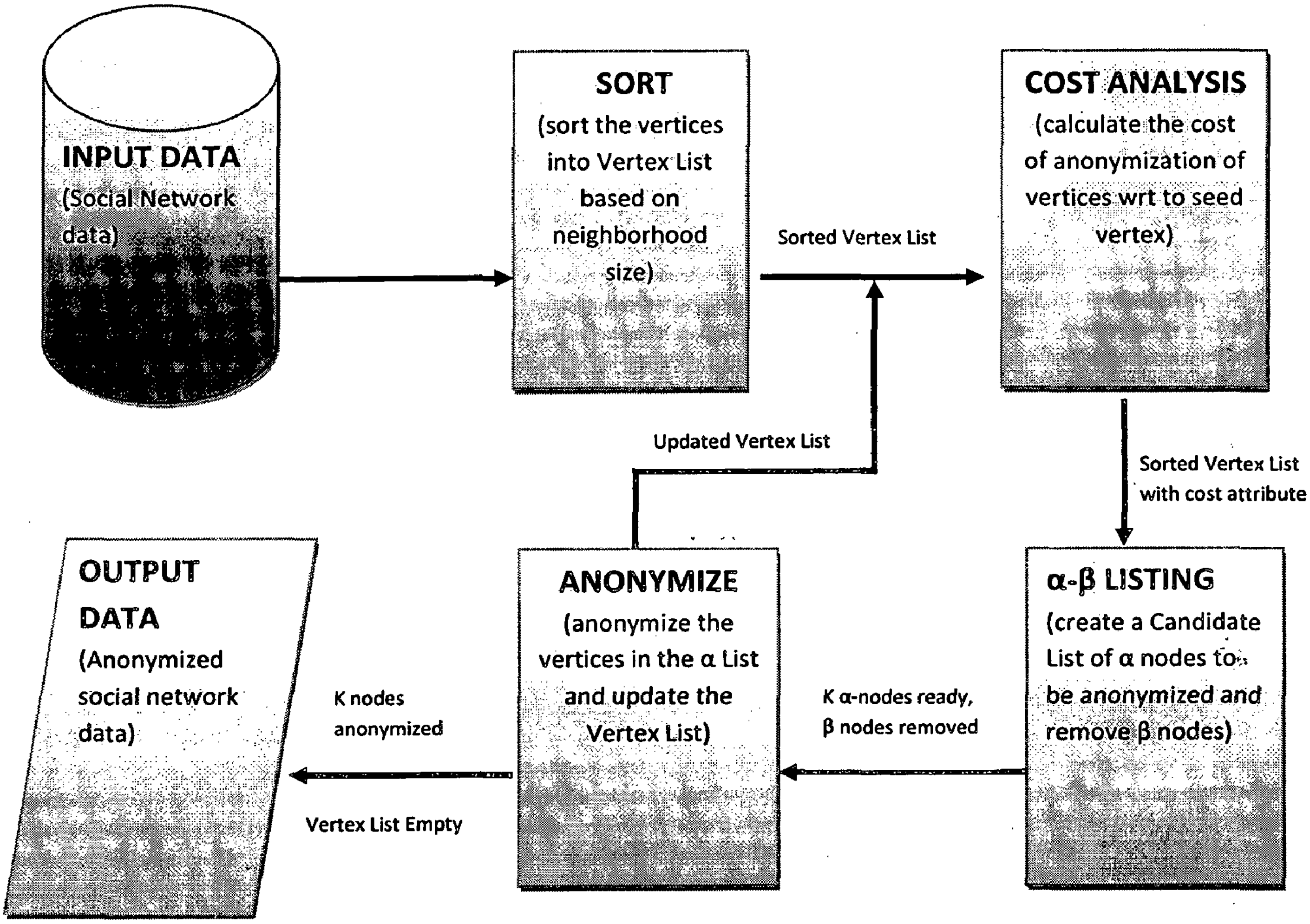

INPUT DATA
(Social Network data)
SORT
(sort the vertices into Vertex List based on neighborhood size)
Sorted Vertex List
COST ANALYSIS
(calculate the cost of anonymization of vertices wrt to seed vertex)
Sorted Vertex List with cost attribute
α-β LISTING
(create a Candidate List of α nodes to be anonymized and remove β nodes)
K α-nodes ready, β nodes removed
ANONYMIZE
(anonymize the vertices in the α List and update the Vertex List)
Updated Vertex List
K nodes anonymized
Vertex List Empty
OUTPUT DATA
(Anonymized social network data)

Fig. 3.1: Proposed Framework for Social Network Anonymization

MODULE 1: INPUT DATA FORMAT

As an input we take an un-anonymized social network graph. The graph is considered to be represented using adjacency matrix. If there exists a connection between two nodes a and b in the social network the corresponding matrix entry is represented as '1', otherwise in the case of no connection it is represented as '0'.

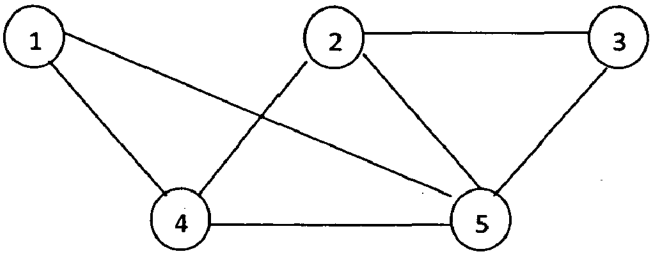


Fig 3.2: Sample social network graph

Table 3.1: Adjacency matrix for the sample graph

	(1)	(2)	(3)	(4)	(5)
(1)	0	0	0	1	1
(2)	0	0	1	1	1
(3)	0	1	0	0	1
(4)	1	1	0	0	1
(5)	1	1	1	1	0

MODULE 2: VERTEX SORT

The vertices in the social network graph need to be arranged in an order which will facilitate efficient and effective anonymization of social network with minimal number of fake edges being added. For that matter, we sort the vertices into the Vertex List in a decreasing order of neighborhood size. We extract the neighborhood graph of a vertex. Based on the total number of vertices and edges, the neighborhood size is determined and the ordering is done.

For the sample social network graph in figure 3.2, the table 3.2 shows the total number of edges in 1-neighborhood of the vertex for all the vertices. Based on this data, we sort the vertices in descending order of neighborhood size as shown in the table 3.3 below.

Table 3.2: 1-neighborhood size

V	Neighborhood size
(1)	3
(2)	5
(3)	3
(4)	5
(5)	7

Table 3.3: Sorted List of vertices

Sorted List
(5)
(2)
(4)
(1)
(3)

MODULE3: COST ANALYSIS

First we select the first vertex in the Vertex List as a seed vertex. Now wrt to this seed vertex we calculate the cost of isomorphism of the vertices in the Vertex List. The cost of isomorphism depends on the number of vertices and edges added to the neighborhood of a two vertices to be isomorphic. Of all the vertices, we need to select k vertices with minimum cost of isomorphism. Different techniques can be applied to avoid calculating cost of isomorphism for all vertices and still extracting k vertices with minimum cost of isomorphism.

MODULE 4: α-β LISTING

In this module we select the node from the Vertex List and lists them as α and β as follows. The first node is selected as seed vertex and listed as α. Then all the nodes in the direct neighborhood of α node are listed as β. All the β nodes are removed from the Vertex List as they are not to be anonymized explicitly. Then the next unlisted node is listed as α while all its direct neighbor are listed as β and removed from the Vertex List. Thus two α nodes are never mutually connected while there may exist a connection between α-β nodes or β-β nodes.

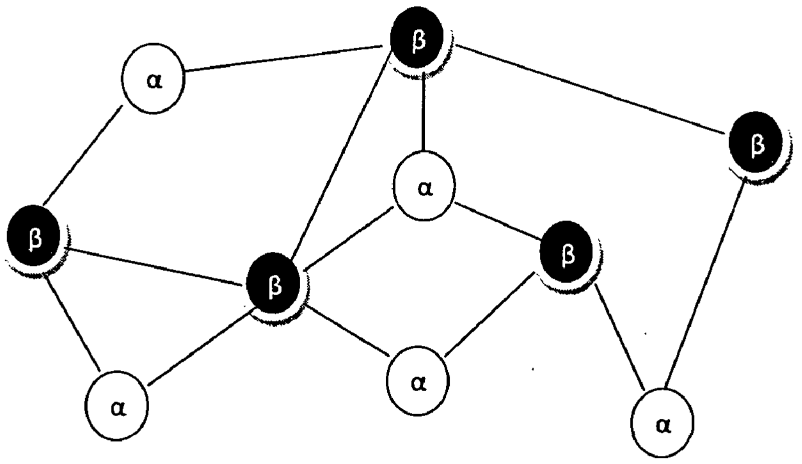


Fig.3.3: α-β structure in a graph

MODULE 5: ANONYMIZATION

To anonymize the social network graph, we take the chunk of k vertices from the sorted Vertex List. Once the group of k-1 vertices (α nodes) is extracted, they are anonymized wrt seed vertex. The group vertices (all α nodes) are anonymized for their 2-neighborhood using anonymization procedure based on adjacency matrix. Once anonymized, the vertices in the group are labeled as anonymous and removed from the Vertex List.

Before exiting, we perform a check on Vertex List to see if it is empty or not. If not we begin the cost analysis of anonymization of remaining vertices in the Vertex List wrt to the seed vertex (first vertex in the Vertex List). And if the Vertex List is empty, we end the process and consider the social network graph to be anonymized.

3.2. Anonymization against neighborhood attacks:

All the nodes are categorized into either α or β. If a node is listed as α, it is k-anonymous to 2-neighborhood i.e. its neighborhood graph upto the depth of 2 nodes is isomorphic to 2-neighborhood graph of (k-1) other nodes. Else if a node is β listed, it may either belong to neighborhood of exactly one α node or certain α nodes. In the former case, it shall be k-anonymous to 1-neighborhood with the respective node of k-anonymous α group. In the latter case, its substructure corresponding to different α-groups shall be k-anonymous up to 1-neighborhood with corresponding nodes of respective k-anonymous α-groups.

Thus all the nodes whether α or β are at least k-anonymous upto 1-neighborhood of graph, which makes the complete social network graph anonymous against neighborhood attacks up to 1-neighborhood.

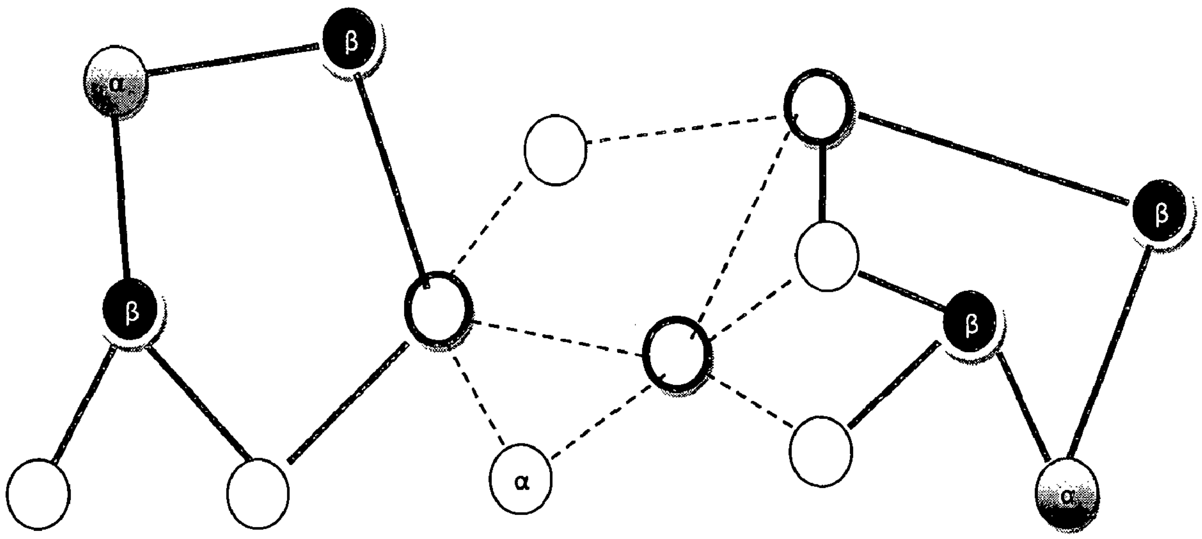


Fig 3.4: Two anonymous α nodes in a sample graph

3.3. Anonymization against friendship attacks:

Friendship attacks are 1-2 attacks i.e. they are edge based attack where edge represents the relationship/friendship between the two mutually connected nodes. For example, F{20,30} attack will detect the two nodes which are mutually connected and have degree 20 and 30.

Thus, based on our α-β categorization of nodes, friendship can only be α-β attack. There cannot be α-α attacks because according to our scheme, two α nodes can never be connected to each other directly, whereas at the end of anonymization all the β-β connections get changed to α-β connections.

VI. ADVANTAGES OF THE PROPOSED SYSTEM

The proposed algorithm is advantageous over the existing one in the following ways:

1. The proposed algorithm can be used to preserve privacy in the published social network against neighborhood and friendship attacks.

2. The proposed algorithm can easily be extended to higher values of d as described in '3' below. The algorithm of Zhou and Pei uses minimum DFS code to get isomorphism checks.

But as d increases the number of possible DFS trees for every component increases exponentially.

3. The proposed algorithm is easier to implement and has a time complexity less than the existing one as isomorphism algorithm uses the adjacency matrix instead of the DFS code and the adjacency matrix is constructed by taking ordering of vertices in groups of descending order.

4. Since the 'd'th power of the adjacency matrix provides the paths of length 'd' between different vertices, we can use it to tackle the adversary knowledge up to d^{th} immediate neighbor.

CHAPTER 4
IMPLEMENTATION DETAILS

In this chapter, we present the implementation details and the results obtained of the solution proposed. To evaluate our anonymization method, both synthetic and real datasets were used.

4.1. System & Software Used:

All the experiments were conducted on a PC computer running the Microsoft Windows 7 Professional Edition operating system, with a 3:0 GHz Core i5 CPU, 8.0 GB main memory, and a 500 GB hard disk. The program was implemented in C/C++ and was compiled using Bloodshed Dev C++ 4.9.9.0.

4.2. Real Dataset Used:

We used a real co-authorship data set from KDD Cup 2003 to examine whether neighborhood attacks may happen in practice. The data set was from the e-print arXiv (arXiv.org), and contains a subset of papers in the high-energy physics section of the arXiv. The LATEX sources of all papers are provided. We extracted author names from the data sources and constructed a co-authorship graph. Each vertex in the graph represents an author, and two vertices are linked by an edge if the two corresponding authors co-authored at least one paper in the data set. There are around 100 vertices and 250 edges in the co-authorship graph and the average number of vertex degrees is about 4. We also ran our tests on synthetically generated random datasets with different average degree to study time performance and method performance for various values of K the anonymization constant.

4.3 Design and Development of System Functions:

4.3.1. Code Design for Anonymization of Two Vertices

void Anonymize (int, int)

The above function takes two integer parameters and anonymizes the neighborhood of the vertices represented by those two integer inputs of the function.

It first extracts the neighborhood of vertex corresponding to first integer, and puts them into a vector. After that it extracts the neighborhood of vertex corresponding to second integer and puts them into another vector.

Code Snippet:

```
            for (int i=0;i<N;i++){
                        if(edge[first][i]){
                              P.push_back(i);
                              sp++;
                        }
             }
```

It then compares the number of vertices in both the vectors and fills additional vectors required wherever to make them equal. It then sorts the vertices in both the list based on the vertex degree. After sorting, it begins edge by edge comparison for all the pair of vertices in both the vector lists and adds the fake edges wherever required to make the neighborhood of the two vertices isomorphic and thus the two vertices are anonymized.

Code Snippet:

```
            for(int i=0;i<sizeq;i++){
                for(int j=0;j<sizeq;j++){
                  if((edge[Q[i]][Q[j]]) && (!edge[P[i]][P[j]])){
                    edge[P[i]][P[j]]=1;
                                         //degree[P[i]]++;
                                         //FEcount--;
                                         cout<<endl<<FEcount;
                    printf(" Edge added from node %d to %d :",P[i],P[j]);
                                         FEcount++;
                                         cout<<FEcount;
                              }
                }}
```

4.3.2. Code Design for Running Time Evaluation:

To evaluate the running time of the program, following code snippet was used:

Code Snippet:

```
#include <time.h>
clock_t t1,t2;
 t1=clock();
//code to be run
 t2=clock();
float diff ((float)t2-(float)t1);
 cout<<"\nClock Ticks: "<<diff;
 float seconds = diff / CLOCKS_PER_SEC;
 cout<<"\nTime in Seconds: "<<seconds;
```

This time function generates different time value each time the program is run for exactly same parameters. To counter-effect the variation, we run the program for few times for same value of parameters and take the average value of the time values during those runs.

4.4. Results and Performance Evaluation:

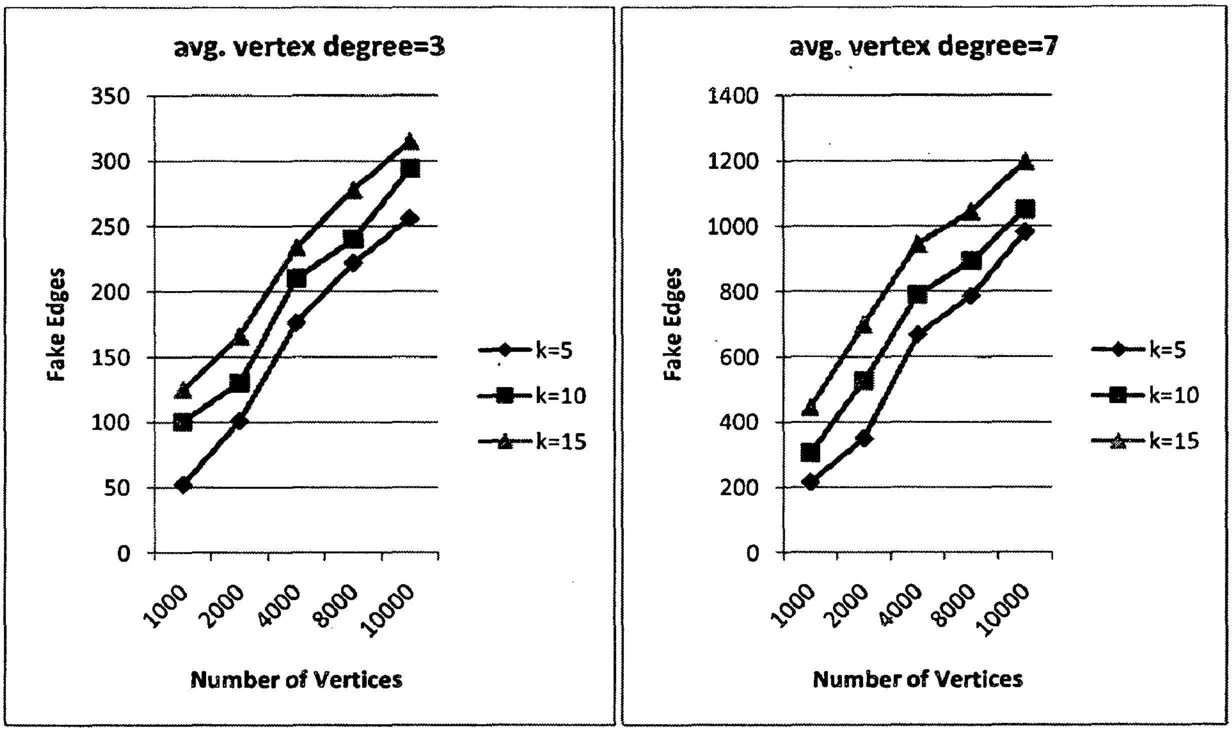


Fig.4.1: Anonymization Cost (Fake Edges) on various synthetic datasets

The program was run for various synthetic datasets. The datasets generated synthetically were categorized based on average vertex degree into two groups with vertex degree 3 and 7. Based on the fixed average vertex degree, datasets were generated of varied sizes of total number of vertices from 1000 to 10000 vertices. On performing the program evaluation on these datasets, the above graphical statistics about fake edges added were obtained.

As expected, the average numbers of fake vertices added were increased with increase in the average vertex degree, because the permutations of various neighborhood patterns increases with increase in average vertex degree. Hence, more number of fake vertices need to be added for increased combinations of neighborhood for k-anonymization. As depicted in the above graph, for the same size of dataset in terms of total number of vertices, the total number of fake vertices added is more for dataset with average vertex degree 7 than with average vertex degree 3.

Also, with increase in the dataset size with same average vertex degree, the numbers of fake vertices added were increased. This behavior can be attributed to the fact that more anonymization need to be done for more number of vertices.

The program running time was also evaluated for various sizes of datasets. As expected, it was observed that running time of the program increases with increase in anonymization cost or dataset size.

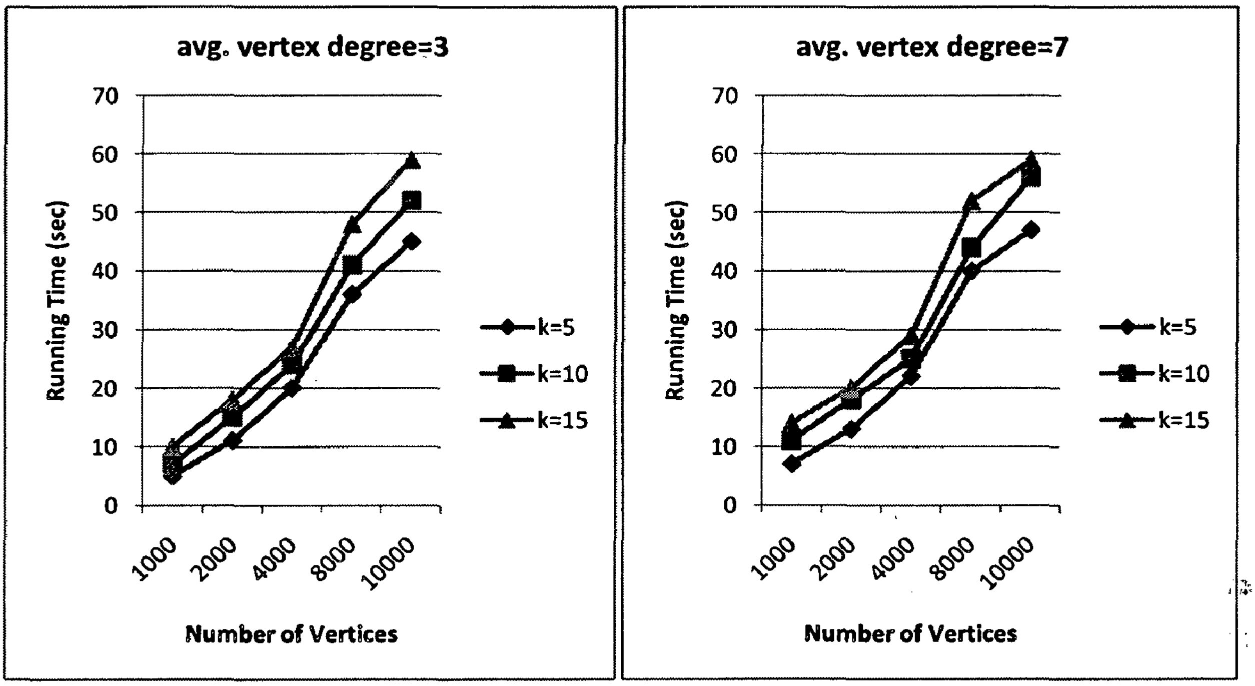


Fig. 4.2: The runtime on various synthetic datasets

To compare the performance of the two anonymization techniques based on adjacency matrix and min-DFS code, we ran the two respective programs on the same dataset. Various datasets of different sizes were generated with the average vertex degree equal to 7. The value of anonymization constant was set to be 5. The results obtained are shown in fig 4.3.

As can be seen in the figure 4.3, the fake edges added in the case of min-DFS technique are less than that of the Adjacency Matrix technique. This can be attributed to the fact our adjacency matrix technique anonymizes the social network for friendship attacks over the already anonymized social network for neighborhood attacks. So at the cost of addition of few more fake edges using our proposed approach, the social network gets anonymized against both neighborhood and friendship attacks. The running time results generated came out as expected. Due to lesser time complexity of algorithm using Adjacency Matrix approach as compared the one using min-DFS approach, the time values for former were lesser than that for the latter one.

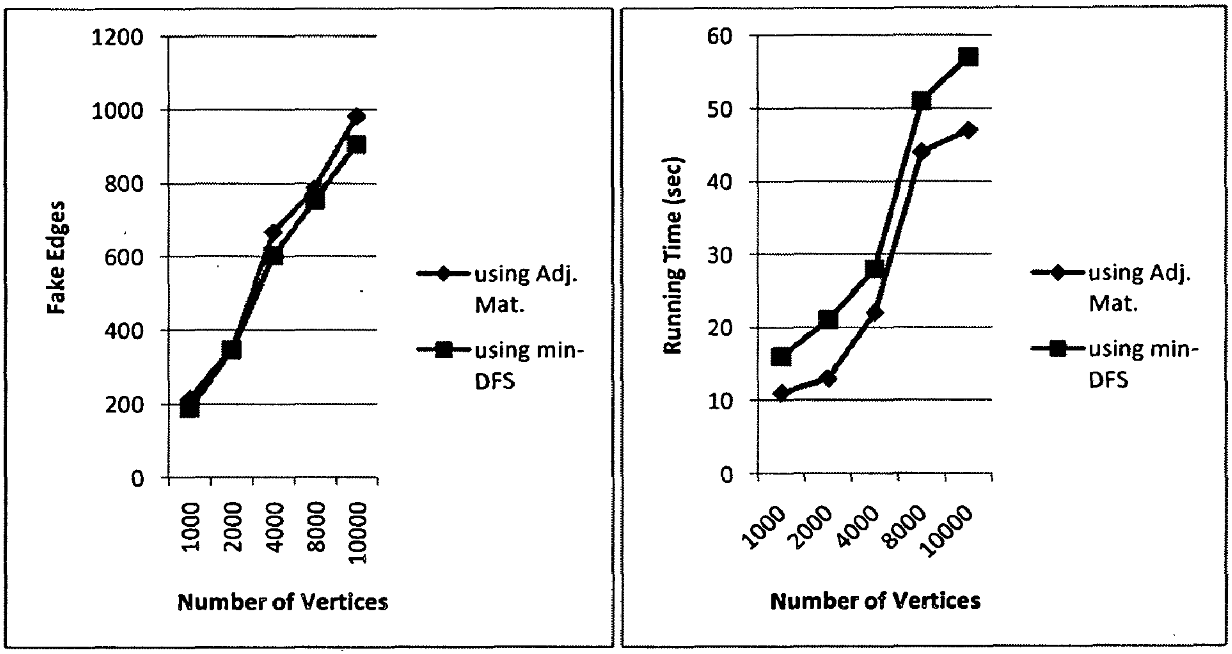


Fig.4.3 Comparison results for Adj. Matrix and min-DFS technique

To test the program on real dataset, we took a sample of the real co-authorship data set from KDD Cup 2003. A sample of around 100 nodes and 250 edges was recovered from the large dataset and results obtained were as shown in the figure below. The fake edges added with increasing K (anonymization constant) showed increasing pattern except at one point for K=15. Also the program running time increased with increase in value of K. The exceptions at some points can be due to the small size of the dataset. The results become more regular as the size of dataset becomes large.

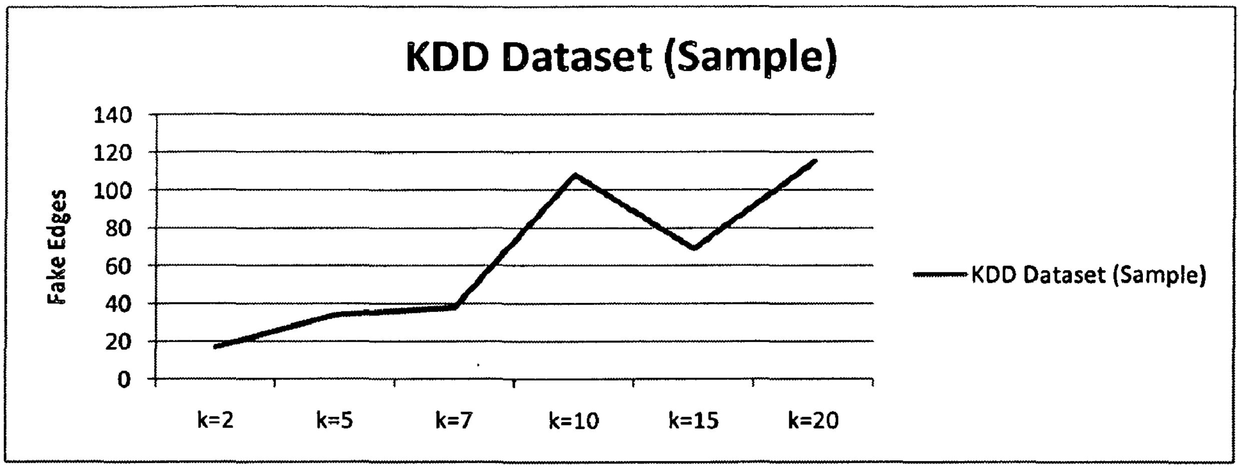


Fig.4.4: Anonymization Cost (Fake Edges) for various values of K (anony. const)

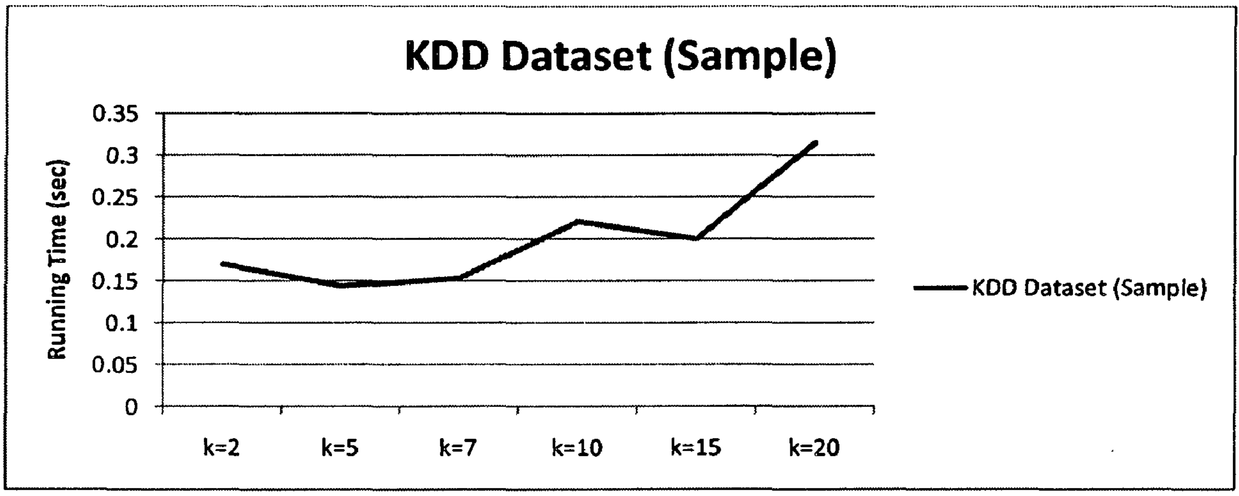


Fig.4.5: The runtime for various values of K (anony. const)

CHAPTER 5
CONCLUSION AND FUTURE WORK

Preserving privacy in publishing social network data has become an important concern. With some local knowledge about individuals in a social network, an adversary may attack the privacy of some victims easily. Thus, several methods have been developed to tackle this problem earlier. But most of these studies on privacy preservation can deal with relational data only and cannot be applied to social network data. An algorithm based on min-DFS code, proposed and studied by Zhou and Pei [10] takes care of neighborhood attacks by using the technique of k-anonymization. However, this algorithm has many limitations. Later Tripathi and Panda[13] improved the above algorithm by using adjacency matrix which was more efficient. Tai, Yu and Yang[14] proposed an algorithm to tackle friendship attacks separately. In this dissertation, we have modified their approach and proposed an algorithm so that the social network can be anonymized against neighborhood and friendship attacks simultaneously. The results obtained were encouraging and this proposed system can be made more robust to make it practically applicable.

5.1 Suggestions for future work

1. The number of fake edges added were individually more than those for anonymization against neighborhood and friendship attacks separately..The algorithm can be made more efficient to improve anonymization performance.

2. The proposed algorithm was tested on smaller datasets and can run only on a single system. Distributed algorithm based on the proposed solution can be developed to be applicable on enormously huge real social network datasets.

3. Similar algorithms can be developed for the cases where the adversary has individual information for greater depths of vertices, on the similar lines.

REFERENCES

[1]. Wassermann, S. and Faust, K., Social Network Analysis: Methods and Applications, Cambridge: Cambridge University Press, 1994.

[2]. Samarati, P. and Sweeney, "L.,Generalizing data to provide anonymity when disclosing information," in Symp. on Principles of Database Systems (PODS), 1998.

[3]. J. Han and M. Kamber, Data Mining: Concepts and Techniques, San Francisco: Morgan Kaufmann Publishers, 2001.

[4]. Sweeney, L., "k-anonymity: a model for protecting privacy", International Journal on uncertainty, Fuzziness and Knowledge-based System, vol. 10, no. 5, pp. 557–570, 2002.

[5]. Machanavajjhala, A., Kifer, D., Gehrke, J and Venkitasubramaniam, "M.: l-diversity: Privacy beyond k-anonymity," in Proc. of the International Conference on Data Engineering (ICDE), 2006.

[6]. D.-W. Wang et al., "Privacy protection in social network data disclosure based on granular computing," in Proc. of the 2006 IEEE Int'l Conference on Fuzzy Systems, pp. 997–1003, 2006.

[7]. Hay, M., Miklau, G., Jensen, D., Weis, P. and Srivastava, "S.: Anonymizing social networks," University of Massachusetts Amherst, Tech. Rep., pp. 07-19, 2007.

[8]. E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationships in graph data," in Proc. of PinKDD' 07, pp. 153-171, 2007.

[9]. L. Backstrom et al., "Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography," in The World Wide Web Conference, 2007.

[10]. Zhou, B. and Pei, J., "Preserving privacy in Social networks against neighborhood Attacks", Simon Fraser University, In Proc. of the International Conference on Data Engineering, pp.506-515, 2008.

[11]. Thompson, B. and Yao, D., "The Union-split Algorithm and Cluster-Based Anonymization of Social Networks", ASIACCS'09, Sydney, NSW, Australia, March 10-12, 2009.

[12]. J. Cheng, A. W.-C. Fu, and J. Liu. "K-isomorphism: privacy-preserving network publication against structural attacks," In Proc. of the ACM SIGMOD, 2010.

[13]. B. K. Tripathy and G. K. Panda, "A New Approach to Manage Security against Neighborhood Attacks in Social Networks," In Proc. of the 2010 Int'l Conference on Advances in Social Networks Analysis and Mining (ASONAM '10). 2010.

[14]. CH Tai, PS Yu, DN Yang. "Privacy-preserving social network publication against friendship attacks," In Proc. of the 17th ACM, 2011.

PUBLICATIONS

[1].Gulshan Budhwani and Durga Toshniwal, "Social Network Anonymization against Neighborhood and Friendship Attacks," In Proc. of the International Conference on Computer Science and Engineering (CSE), Guwahati, India, June, 2012.