An Algorithm to achieve k-anonymity and l-diversity anonymisation in Social Networks

B. K. Tripathy
SCSE, VIT University
Vellore 632014, Tamil Nadu, India.
e-mail: tripathybk@vit.ac.in

Anirban Mitra
Department of CSE, MITS,
Rayagada 765017, Odisha, India.
e-mail: mitra.anirban@gmail.com

***Abstract*—The development of several popular social networks in recent days and publication of social network data has led to the danger of disclosure of sensitive information of individuals. This necessitated the preservation of privacy before the publication of such data. There are several algorithms developed to preserve privacy in micro data. But these algorithms cannot be applied directly as in social networks the nodes have structural properties along with their labels. k-anonymity and l-diversity are efficient tools to anonymise micro data. So efforts have been made to find out similar algorithms to handle social network anonymisation. In this paper we propose an algorithm which can be used to achieve k-anonymity and l-diversity in social network anonymisation. This algorithm is based upon some existing algorithms developed in this direction.**

Keywords-social networks, k-anonymity, l-diversity, anonymisation

I. INTRODUCTION

A social network describes entities and connections between them. Social network analysis is concerned with uncovering patterns in the connections between entities. For this the data associated with such networks need to be published. But the major problem faced in such publication is the disclosure of sensitive information of the participants, which is highly undesirable. So, before the publication of these data sufficient care must be taken to hide the sensitive details. To achieve this we need to maintain balance between privacy and utility in data publishing. Several algorithms in this direction have been developed for micro data. More effective of these approaches are based upon the k-anonymity, the ℓ -diversity and the t-closeness techniques. However, such algorithms have the common assumption that the records are independent of each other and can be anonymised more or less independently. In contrast to this assumption, social network are more challenging to anonymise due to the difficulty in modeling of background knowledge of adversaries, linked relationship among nodes, difficulty in measuring information loss due to anonymisation and removal or addition of edges may affect the properties of the network.

In order to attack the privacy of a target individual in the original network, that is to analyze the released anonymised network and re-identify the vertex, an adversary needs some background knowledge. Among the types of information about a target vertex an adversary may collect how the neighbours are connected to each other. Generally, d-neighbours of a target vertex are considered, That is the information about the vertices within distance d from the target vertex can be used. Such type of attack by adversaries is called as neighbourhood attack. Zhou and Pei [11] proposed an anonymisation technique for social networks to prevent the neighbourhood attacks. Their technique depends upon the DFS codes and minimum DFS codes. They had restricted themselves to the case when d = 1. However, Tripathy et al [5, 7] have modified the algorithm suitably to make it more efficient and also handle the case when d > 1.

The k-anonymity model emphasizes upon the existence of a minimum of k vertices in the anonymised network which are similar to each other. That is in the anonymised network a node cannot be re-identified with confidence more than 1/k.

Machanavajjhala et al [2] introduced three types of ℓ -diversities. Out of these the distinct ℓ -diversity has been tackled successfully so far for micro data anonymisation and a three phase algorithm has been developed by Tripathy et al [8] in this direction. The other types of ℓ -diversities are yet to be handled in any algorithm so far. Zhou and Pei [12] have developed an ℓ -diversity algorithm for the anonymisation of social networks, by modifying a k-anonymity algorithm developed by them earlier.

In this paper, we provide an alternate ℓ -diversity algorithm for anonymisation of social networks, which depends upon the algorithm for social network anonymisation developed by Tripathy et al [5, 7] and the three phase algorithm developed by Tripathy et al [8] for anonymisation of data satisfying the l-diversity criterion.

Due to constraint of space, we shall not provide the details of experimental analysis of this proposed algorithm. In the next section we provide some definitions

978-1-4673-4794-5/12/$31.00 ©2012 IEEE

to be used in the development of the algorithm of this paper.

II. DEFINITION AND NOTATIONS

We provide below the definitions of some terminologies to be used in the rest of the paper.

.

Definition 2.1: Modeling a social network

A social network is modeled as a simple graph $G = (V, E, L, \zeta)$, where V is a set of edges, where V is a set of vertices, $E \subseteq V \times V$ is a set of edges, L is a set of labels and a labeling function $\zeta : V \to L$ assigns each vertex a label. For a graph G, V(G), E(G), L_G, ζ_G are the set of vertices, the set of edges, the set of labels, and the labeling function in G, respectively.

Definition 2.2: Label Hierarchy

The items in the label set L of nodes in the graph of a social network forms a hierarchy.

For example, if the occupations are used as labels of vertices in a social network, L contains not only the specific occupations such as dentist, general physician, optometrist, high school teacher, and primary school teacher but also general categories like medical doctor, teacher, and professional.

It is assumed that there exists a meta symbol $* \in L$, which is the most general category generalizing all labels.

Definition 2.3: Neighborhood of a vertex and Neighborhood component

In a social network G, the neighborhood of u $\in V(G)$ is the induced sub graph of the neighbors of u, denoted by $Neighbor_G(u) = G(N_u)$, where $N_u = \{v / (u, v) \in E(G)\}$. The components of the neighborhood graph of a vertex are the neighborhood components. In a social network G, a sub graph C of G is a neighborhood component of $u \in V(G)$ if C is a maximal connected sub graph in $Neighbor_G(u)$.

Definition 2.4: d-Neighborhood

The d-Neighborhood graph of a vertex u includes all the vertices that are within the distances'd' from the vertex u. The distance is measured as the number of edges in the shortest path connecting the two nodes.

An adversary may attack the privacy using the neighborhoods. For a social network G, suppose an adversary knows $Neighbor_G(u)$ for a vertex $u \in V(G)$. If $Neighbor_G(u)$ has k instances in G' where G' is an anonymisation of G, then u can be re-identified in G' with confidence 1/k.

Similar to the philosophy of k-anonymity model [3, 4], to protect the privacy of vertices sufficiently, we want to keep the re-identification confidence lower than a threshold. Let k be a positive integer. For a vertex $u \in V(G)$, u is k-anonymous in anonymisation G' if there are at least (k – 1) other vertices $v_1, v_2, \ldots v_{k-1} \in V(G)$ such that $Neighbor_{G'}(A(u)), Neighbor_{G'}(A(v_1(), \ldots Neighbor_{G'}(A(v_{k-1}))$ are isomorphic. G' is k-anonymous if every vertex in G is k-anonymous in G'.

III. ALGORITHMS AND COMPUTATIONAL PROCEDURE

To determine the isomorphism of neighbourhood graphs of two vertices the DFS code and component technique was used by Zhou and Pei [11]. However, a more efficient technique was developed in Tripathy et. Al. where a improved brute force isomorphism algorithm was developed and was established to be efficient under the circumstances. For this purpose, the adjacency matrices of the components of the neighbourhoods of the nodes were considered.

The advantage of this approach is its extensibility to handle d-neighbourhood attacks by using the computation of the dth power of the adjacency matrices. Also, due the following two properties are helpful to see that the computation procedure does not have much complexity.

Property 3.1:Vertex degree in power law distribution

The degrees of vertices in a large social network follow a power law distribution that is only a small number of vertices have a high degree. Processing the higher degree vertices first can keep the information loss about those vertices low. Often, there are many vertices of lower degrees. It is relatively easier to anonymise those lower degree vertices and retain high quality.

Property 3.2:Small-world phenomenon [14]

This phenomenon states that large practical social networks often have surprisingly small average diameters.

We present the isomorphism developed in [7,10] below:

3.1 Revised Brute Force Graph-isomorphism Testing

Two graphs are isomorphic if it is possible to order their respective vertex-sets so that their adjacency matrices are identical.

Input: Graphs G and H.

Output: YES or NO, according to whether G is isomorphic to H or not.

1. Let V_G and V_H denote the set of vertices of the graphs G and H respectively.
2. If $|V_G| \neq |V_H|$ Return NO.
3. Else, put the vertices in G and H in the descending orders of their degrees.
4. If degree sequences are not equal return NO.
5. Write the adjacency matrices A_G and A_H of G and H respectively with respect to the ordering of their vertices as above.
6. Let the number of vertices of order 'i', i = 1, 2 … k, in the graphs G and H be Gn_i and Hn_i.
7. For j = k,…1
8. If $Gn_i \neq Hn_i$ for some I then return NO.
9. Else, let $[A_G]_j$ and $[A_H]_j$ denote respectively the sub matrices of A_G and A_H corresponding to the vertices of order j respectively.
10. For a particular ordering of $[A_G]_j$ write $[A_H]_j$ in that order
11. If $[A_G]_j \neq [A_H]_j$ then return NO.
12. Else, return YES.

The neighborhood graphs of all the vertices are separated into their components and are represented in the form of adjacency matrices. The adjacency matrix is constructed in the decreasing order of the vertices and their label in the component. When two or more vertices have the same degree the ordering is done according to decreasing label.

Two components with the same degree and having same adjacency matrices are isomorphic according to their structure. Next, if the labels also match then they are isomorphic. Else the labels are generalized to their parent label. The similarity between the components with different number of vertices is done by comparing the first sub matrices of the adjacency matrices of the components with highest number of vertices. In non-matching cases, vertices or edges can be added for anonymisation or making them isomorphic.

For the anonymisation procedure and hence the network anonymisation procedure we refer to Tripathy [7]. Also, we mention that a comparative study of efficiency of this approach and the existing approach due to Zhou and Pei [11] is provided in [5, 7].

3.1.1 ℓ -diversity in social networks

For the discussion on k-anonymity and l-diversity in social network context we use the following generalised definition of a social network model put forth in [7].

Definition 3.1.1

A social network is modeled as a simple undirected graph G = (N, E), where N is the set of nodes and $E \subseteq N \times N$ is the set of edges. Each node represents an individual entity. Each edge represents a relationship between two entities.

The set of nodes, N, described by a set of attributes that are classified into the following three categories:

$I_1, I_2, ...I_m$ are identifier attributes such as the name and SSN that can be used to identify an entity.

$Q_1, Q_2, ...Q_q$ are quasi-identifier attributes such as zip-code and sex that may be known by an intruder.

$S_1, S_2, ...S_r$ are confidential or sensitive attributes such as diagnosis and income that are assumed to be unknown to an intruder.

Binary relationships are used in the model and all relationships are considered as being of the same type and as a result, the relations are represented as undirected edges. The graph structure may be known to an intruder and used by matching it with known external structural information, therefore serving in privacy attacks that might lead to identity and/or attribute disclosure.

As identified in Machanavajjhala et al [2], even a k-anonymised table can leak enough of sensitive information unless there is diversity in the values of sensitive attributes. An intruder can use background knowledge to infer the sensitive information regarding a respondent. However, ℓ -diversity takes care of this problem. A fast and efficient three phase algorithm is developed by Tripathy et al [8] to take care of ℓ -diversity for micro data. In this paper, we shall outline how this algorithm for use in anonymisation of social networks. Also, we shall provide an improved second stage of this algorithm.

3.2 The improved three phase ℓ –diversity algorithm

The three phase algorithm developed and tested in [8] is an extended version of the OKA algorithm [11]. In fact, OKA had two phases (clustering and adjustment phases) only so that it takes care of k-anonymity. A third phase, the –diversity phase was added in [8], besides improving

the first two phases to make it more efficient. Here, we provide a further improvement of the second phase. The algorithm is as follows:

3.2.1. The Modified Clustering Phase Algorithm

Input: A set T of n records; the value k for k-anonymity and the value l for l-diversity.

Output: A partition P = {P1, P2 …PK}

1. Let r be the first record in T
2. Order {Pi } according to their distances from r;
3. Let i=1;
4. Flag = 0;
5. While ((i < K) and(Flag= 0))
6. Let $s(P_i)$ be the set of distinct sensitive attribute values of the tuples in Pi;
7. Let $s(r)$ be the sensitive attribute value of r;
8. If (($|P_i| < k$) or (($s(r) \notin s(P_i)$) and($|s(P_i)| < l$)) then add r to Pi ;
9. Update centroid of Pi;
10. Flag = 1;
11. Else i:= i+1;
12. End of while
13. If(Flag = 0) add r to the nearest cluster // decided by the cost (centroid of the cluster, r)
14. Let T:=T\{r};
15. End of while

3.2.2. The Improved Adjustment Phase Algorithm

In this stage we modify the algorithm proposed in [8] for the second stage order to make it more efficient. We present the modified algorithm as follows:

Input: A partition $P = \{P_1, P_2, \ldots, P_K\}$ of T.

Output: An adjusted partition $P = \{P_1, P_2 \ldots\ldots P_M\}$ of T

1. Let S be the set of clusters $P \in P$ with $|P| \le k/2$.
2. Let U be the set of clusters in P such that $k/2 < |P| < k$.
3. Let U be the set of clusters in P such that $|P| \ge k$.
4. Flag = 0
5. While ((S¹f) and (Flag=0)) do
6. Select P from S
7. While ((U ≠ f) and (P ≠ f)) do
8. Select a cluster Q from U closest to P;// decided by cost (centroids of the clusters)
9. Add min {k-|Q|,|P|} elements W from P to Q;
10. If($|P| \ge k - |Q|$)
11. Let U = U - {Q} , V=VU{Q} and P = P-W.
12. End of while
13. If(U = f)
14. Flag = 1.
15. S = S- {P}.
16. End of while
17. If (Flag = 0)
18. Let S' = U{P : P ∈ S}.
19. While (S' ≠ ϕ)
20. Select $r \in S'$.
21. Add r to the closest cluster in V// decided by the cost (centroid of cluster V, r)
22. End of while
23. Else
24. While (U ≠ ϕ)
25. Select $Q \in U$.
26. While ($|Q| < k$)
27. Select P∈ V with |P| > k
28. Sort records in P by distance to centroid of P; // decided by the cost(centroid, record)
29. While (|P | > k)
30. r∈ P is the record farthest from centroid of P ;// highest cost(r, centroid P)
31. Let P = P \{r}; Q = $Q \cup \{r\}$;
32. End of While;
33. End of While;
34. U=U-{Q};V=VU{Q}.
35. End of While;
36. End else.

3.2.3. The ℓ - Diversity Phase Algorithm

Input: Clusters formed after adjustment stage (M in number)

Output: Clusters satisfying ℓ -diversity

1. Let P be the matrix of frequencies of attribute values, whose columns correspond to the clusters and rows correspond to the different attribute value tuples in the domain of the set of p sensitive attributes. The last row contains the diversity values d_i for the clusters (equal to the number of non-zero values in the corresponding column). The entries in P other than those in the last row contain frequencies of attribute value tuples in the clusters.
2. Order the columns in P according to the ascending order of the diversity values.
3. Let $q = \max\{i : d_i < \ell\}$.

4. For each cluster
   C_i such that $1 \le i \le q$,
   compare with cluster C_j, j = q+1… m.
5. F = {the sensitive attribute values which are in C_j but not in C_i and have frequency greater than 1}.
   Find $m_i = \min\{(l - d_i), |F|\}$ of them which are closest to the tuples in C_i.
6. Interchange mi tuples between C_i (Those tuples with sensitive values > 1) and C_j s.
7. Increment the diversity of C_i by m_i.
8. Continue the process till the diversity of all C_i is 'l' or no cluster is left in $\{C_j, q+1 \le j \le m\}$ for comparison.
9. Let S = $\{C_i : \text{diversity of } C_i < l\}$.
10. If $If\ |S| > 1$ then
11. C be the first element of S. Compare it with other clusters in S
12. Perform steps 5 to 8;
13. Else merge the element in S with the nearest of the clusters with diversity l or more obtained above.

3.2.4. Anonymising a social network to achieve ℓ -diversity (Modified from [13])

We replace the partition (VertexList) function in the algorithm provided by Zhou and Pei [13] to get the ℓ -diversity of social network.

Input: A social network G = (V, E), the anonymisation requirement parameter k, the cost function parameters α, β *and* γ ;

Output: An anonymised graph G'

1. Initialize G' = G;
2. Mark $v_i \in V(G)$ as "unanonymized";
3. Sort $v_i \in V(G)$ as Vertex List in neighborhood size descending order;
4. WHILE (Vertex List $\neq \phi$) DO
5. Let Seed Vertex = VertexList.head () and remove it from VertexList;
6. FOR each $v_i \in VertexList$ DO
7. Calculate Cost (Seed Vertex, v_i) using the anonymisation method for two vertices;
8. END FOR
9. IF ($VertexList.size() \ge 2\ell - 1$) then
10. Let CandidateSet = Partition (VertexList);
11. ELSE
12. Let CandidateSet contain the remaining unanonymized vertices;
13. Suppose Candidate Set= $\{u_1, ..., u_m\}$, anonymise Neighbor (Seed Vertex) and Neighbor(u_1)
14. FOR j = 2 to m DO
15. Anonymise Neighbor (u_j) and {Neighbor(SeedVertex),Neighbor(u_1),…Neighbor(u_{j-1})}, mark them as "anonymised";
16. Update VertexList;
17. END FOR
18. END WHILE

3.2.5 Algorithm for Partition of vertex list

Input: VertexList

Output: Partition (VertexList)

1. Cluster-list = Apply Clustering-Phase-Algorithm (VertexList);
2. Adjusted-Cluster-list = Apply Adjustment-Phase-Algorithm (Cluster-list);
3. L-diverse-Cluster-list = Apply l-diverse-phase-Algorithm (Adjusted-Cluster-list);
4. Partition (VertexList)= l-diverse-Cluster-list;

Note 3.1: We have modified algorithm 3.2.1 as the initial vertex, which is called as the seed vertex in algorithm 3.2.4 has already been selected. Also, the centroids in the clusters in the algorithms are nothing but the seed vertices.

IV. ANALYSIS

In a social network, we can have three types of anonymisations.

The first one is the anonymisation of the nodes. In this type of process, every node has a set of attribute attached to it. So that after anonymisation, even if a node is identified somehow, its sensitive attribute values will not be disclosed. The anonymisation algorithms developed for relational databases can be used for this purpose.

The second one is to anonymise the network structure. So, the nodes cannot be identified uniquely, because of their structural similarity. There are several subcases of this procedure and several algorithms have been developed.

The best approach is to follow a combination of both the above cases. In this case we have protection against both the node identification as well as nondisclosure of the sensitive attribute values of nodes. The algorithm developed by Zhou and Pei[13] comes under this category. Not only this but also they have taken care of ℓ -diversity anonymisation in the first case, which is better the usual k-anonymity. Our algorithm proposed here also comes under

this category. However, we have modified technique so that it can be more efficient. Here are the advantages:

1. As discussed our algorithm works for higher degree of distance d >1 as far as structural similarity of nodes are considered. This is because of the incidence matrix approach and the revised brute force algorithm. The revised brute force algorithm is practically helped by the two laws for social networks, property 3.1 and property 3.2. Also, we have used the addition of edges in order to make structure of nodes similar. By this process does not lose any additional information than the loss due to merging of similar nodes. Of course, some redundant information is added. The efficiency of this approach has been established in [5, 7].
2. The step by step computation of the partition takes care of the cluster sizes so as to achieve k-anonymity. Also, the ℓ -diversity is taken care efficiently. This process is an extension of the 2-phase OKA algorithm [ ] and we have provided here the latest improved 3-phase algorithm with some major changes in the 2nd phase. The previous version of this algorithm has been implemented and its efficiency has been compared with other algorithms in this direction by Tripathy et al [8].
3. One more advantage of this algorithm is that it can be extended to a variant of multi-sensitive ℓ -diversity of attributes as illustrated by Tripathy et al [9]. However, the definition of p-sensitivity used in this algorithm is lighter than the one introduced in [2]. But, it is better than the ℓ – diversity of individual sensitive attributes considered by Zhou et al [13].
4. The anonymisation of nodes by taking the components and using the extended brute force graph isomorphism algorithm has less complexity because of the properties 3.1 and 3.2.

The algorithms have been simulated by taking small size social networks represented through graphs. However, the modified implementation (due to changes in the first two phases of the three phase algorithm) from [8] and its combination with the code from [5] have been used. The implementation of the entire algorithm has been performed successfully and found to provide good results on example networks. We avoid presenting the details of the example network and the results due to space constraints.

V. CONCLUSION AND FUTURE WORKS

In this paper we proposed an algorithm which follows k-anonymity, ℓ -diversity properties during anonymisation and it can also be extended to handle a variant of multi-sensitive attributes during anonymisation process. This algorithm is modified suitably from their corresponding algorithm for micro data and also depends upon some modified algorithms developed for anonymisation against neighbourhood attack. The algorithm still needs some improvements in order to reduce the complexity in order that it can be applied to large social networks.

The p-sensitivity problem as mentioned by Machanavajjhala et al [2] is yet to be handled so far even in the relational database case. Only the distinct ℓ -diversity has been considered and used so far. The other types of ℓ -diversities are yet to be handled. Besides these the extensions of ℓ -diversity (like t-closeness) need to be extended. Some more problems have been enumerated in Tripathy[10].

VI. REFERENCES

[1] Gross, J. and Yellen, J.: graph Theory, CRC Press, (2006)

[2] Machanavajjhala, A., Gehrke, J., Kifer, D. and Venkitasubramaniam, M.: l-diversity: Privacy beyond k-anonymity, In Proc. 22nd Intl. Conf. Data Engg.. (ICDE), (2006),24.

[3] Sweeney, L.: k-anonymity: A model for protecting privacy, International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5), (2002), pp.557-570.

[4] Samarati, P. and Sweeny, L.: generalizing data to provide anonymity when disclosing information, in PODS'98, (1998)

[5] Tripathy, B.K., Janaki, L. and Jain Neha: Security against Neighborhood Attacks in Social Networks, Proceedings National Conference on recent trends in soft computing, Bangalore, (2009), pp.216-223.

[6] Tripathy, B.K., Panda, G.K. and Kumaran, K.: A Fast l -Diversity Anonymisation Algorithm, Proc. Of the third International Conference on Computer Modeling and Simulation (ICCMS 2011), Mumbai, 7-9 January, (2011),pp.V2-648- 652.

[7] Tripathy, B.K., Panda, G.K.: A new approach to Manage Security against Neighborhood Attacks in Social Networks, Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining, DOI 10.1109/ASONAM.2010.69, IEEE Computer Society,(2010), Denmark, pp.264 -269.

[8] Tripathy, B.K., Panda, G.K. and Kumaran, K.: An Improved l -Diversity Anonymisation Algorithm, In: proceedings of the Springer international conference, ICIP2011, Bangalore, India, August-2011.

[9] Tripathy, B.K., Maity, A., Ranajit, B. and Chowdhuri, D.: A fast p-sensitive l-diversity Anonymisation algorithm, Proceedings of the RAICS IEEE conference, Kerala, Sept.21-23, (2011), pp.741 – 744.

[10] Tripathy, B.K.: Anonymisation of social Networks and Rough Set Approach, In: A. Abraham (ed.), Computational Social networks: Security and Privacy, Springer Verlag London, (2012), pp.269-309.

[11] Zhou, B and Pei, J.: Preserving privacy in social networks against neighbourhood attacks, Simon Fraser University, In: proceedings of the 24th IEEE International Conference on Data engineering (ICDE '08), IEEE computer society, Cancun, Mexico, (2008), pp.506 -515.

[12] Zhou, B., Pei, J. and Luk, W.S.: A brief survey on anonymisation techniques for privacy preserving publishing of social network data, SIGKDD Explorations, 10(2), (2008), pp.12-22.

[13] Zhou, B. and Pei, J.: The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighbourhood attacks, Knowledge and Information Systems, July 2011, Vol. 28, issue 1, (2011), pp.47-77.

[14] Wasserman, S. and Faust, K.: Social Network Analysis, Cambridge University Press, Cambridge/New York (1994)