Security of User’s Private Attributes

Law Kumar Singh
Department of Computer Science & Engineering
Hindustan College Of Science And Technology
Mathura, India

Priyanka Gavhane
Department of Computer Science & Engineering
Hindustan College Of Science And Technology
Mathura, India

***Abstract*—Social networking sites contain large number of data about users which needed to be safe, but till now user private information is still prone to revelation threat. Here we categorized data as Private (which user not want to share with other) and Non-private (information that can be share with other). The challenge is to implement methods to publish these data in a form that affords benefit without threaten secrecy. Research on this matter till now has proposed various secrecy models. Social networks are represents as graphs in which users are nodes. The threat definitions and the security mechanisms may influence structural properties of the graph. This paper is derived for the need of a more protected system for user information attributes. Here we suggest a secrecy security plan that avoids the revelation of user’s status but also the revelation of selected attributes in profiles of users. An individual can choose which information he wants to secure. Networks are showed as diagram in which users are nodes and its information is tags. Tags are represented either as private or as non-private. We treat node tags both as background knowledge an outside person (intruder) may acknowledge, and as private information that has to be protected. However, we suggest an algorithm in which graph is represented in such a manner that an intruder having some information about a node's sidewise locality cannot predict about the private tags of user. So we transform graph into such a form in which nodes are alike. This algorithm preserves private information efficiently. It offers stronger secrecy guarantees. In this paper we also demonstrate the comparison betweenFNA (False Node Algorithm) &Extended False Node Algorithm (EFNA) algorithm.**

I. INTRODUCTION

The publication of social network data brings a secrecy threat for their users. Private information about should be protected. The challenge is to derive methods to reveal users information in a form that doesn’t compromising secrecy. Online social networking websites make opportunities for users to share their information, keep in touch with others, and maintain professional circles. However such data is prone to be get attacked by intruder.

Till now various secrecy models with the corresponding security mechanisms that prevent both inadvertent private information leakage and attacks by malicious adversaries. These early secrecy models are mostly concerned with identity and link revelation. The user’s networks are showed as graphs in which users are nodes and the links between users are represented as edges. This paper is inspired by the identification of finer grain and more personalized secrecy mechanism. The networks are showed as graphs in which users are nodes and attributes are tags1. Tags are denoted either as private or as non-private. Fig 1 is a described graph representing a small view of such a social network. Each node in the graph represents a user, and the edge between two nodes represents the link between the two persons (friends). Each letter represents a city (location attribute of user) name as a tag for each node. Some peoples (users) do not mind their residence being known by the others, but some do. In such a case, the secrecy of their should be protected at data release. Therefore the locations are either private (tags are in italic in Fig 1) or non-private. Here we also maintain the information of the sidewise locality information for each node that has private tag.

The secrecy issue arises from the revelation of private tags. One might suggest that such tags should be simply deleted. Still, such a solution would present an incomplete view of the network and may hide interesting statistical information that does not threaten secrecy. A more sophisticated approach consists in releasing information about private tags, while ensuring that the identities of users are protected from secrecy threats. We consider such threats as sidewise locality attack, in which an intruder finds out private information based on prior knowledge of the number of neighbors of a target node and the tags of these neighbors.

For example, if an intruder knows that a user has 3 friends A(location is Agra), B (location is Bangalore),C(location is Chennai) then intruder can conclude that the user is in H (location is Hyderabad). We present secrecy security algorithms that allow for graph data to be published in a form such that an intruder cannot safely conclude the identity and private tags of users. We consider the case in which the intruder having both structural knowledge and tag information. The algorithms that we propose convert the original graph into a graph in which any node with a private tag is alike from at least ℓ-1 other nodes. The probability to conclude that any node has a certain private tag (we call such nodes private nodes) is no larger than ℓ/l. For this purpose we design the probability to conclude that any node has a certain private tag (we call such nodes private nodes) is no larger than ℓ - miscellany -like model, where we treat node tags as both part of an intruder's background knowledge and as private information that has to be protected. The algorithms are designed to provide secrecy security while losing as little information and preserving as much utility as possible. We evaluate degree to which the algorithms maintain the original graph's structure and properties.

II. PREVIOUS WORK

This imperative incognito technique in the ambience of micro and network data consists in deleting identification. This core technique has quickly been identified as one which through which secrecy cannot be protected. Backstrom et al., [4], show that core incognito is insufficient as the structure of the modified graph may exhibit the identity of the individuals corresponding to the nodes. Similarly for micro-data, Sweeney et al. propose k-anonymity [5] to side step possible identity revelation in credulously incognito micro data. ℓ-Miscellany is proposed [1] in order to prevent attribute revelation in advance. Cheng et al. [7] give a discursive solution to avoid every kind of attack, including identity revelation, attributes revelation and link revelation. They suggested k-isomorphism, a property with which the graph consists of k pairwise isomorphic subgraphs.

However, as we partitioned the graph along with compelling exaggerationbrings severe change to the graph properties. They also perturb the graphs by adding nodes, adding edges and deleting edges. Ying et al. [5] study on the randomization approaches for link revelation, while [6], they study the consequences of random edge addition, deletion and switching on graph spectrum properties. They look over the incognito level based on a-posteriori belief, which Bonchi et al. [9] believe is not sufficient,Zheleva and Getoor [10] consider graphs containing both private and non-private edges. They assume that the adversaries predict private edges based on the observed non-private edges..Bhagat et al. [8] nameless the described bipartite graphs by grouping similar nodes or edges.Node tags and adding edges. They accomplish a k-anonymity secrecy constraint on the graph, each node of which is assured to have the same immediate sidewise locality structure with other k-1 nodes. [11], they improve the secrecy assured by k-anonymity with the idea of ℓ-miscellany, to protect tags on nodes as well. Yuan et al. [12] tried to be more practical by taking in to consideration users' different secrecy concerns. They divide secrecy obligation into three levels, and suggest methods to derive tags and modify structure corresponding to every secrecy demand. Nevertheless, neither Zhou and Pei, nor Yuan et al. consider tags as a part of the background knowledge. Moreover, as with the context of micro data, a graph that satisfies a k-anonymity secrecy guarantee may still leak out private information regarding its tags [13].

III. PROBLEM STATEMENT

Here we take a network which is showed as G(N,E,L, L_o, F), where N is a set of nodes, E is s set of edges, L_o is a set of private tags, and L is a set of non-private tags. F maps nodes to their tags, $F : V \rightarrow L_o \cup L$. We have propose a secrecy model, ℓ -private-tag- miscellany; in this model, we treat node tags both as part of an intruder's background knowledge, and as private information that has to be protected. These concepts are clarified by the following definitions:

Statement 1.The sidewise locality information of node n comprises the degree of n and the tags of n's neighbors.

Statement 2.(ℓ -private-tag-miscellany) For each node n that associates with a private tag, there must be at least ` ℓ-1 other nodes with the same sidewise locality information, but attached with different private tags.

In Example below, nodes 2, 3 and 7 have private tags. The sidewise locality information of node 1, includes its degree, which is 5, and the tags on nodes 2, 7, 9, 10 and 11 which are Q, S, W, U and V respectively. For node 2, the sidewise locality information includes degree 4 and the tags on nodes 1, 7, 3 and 8 which are P, S, R and T. The graph in Fig 2 satisfies 2-private-tag-miscellany; that is because, in this graph, nodes 1 and 3 are alike, having five neighbors with tag Q, S , U, V, W.

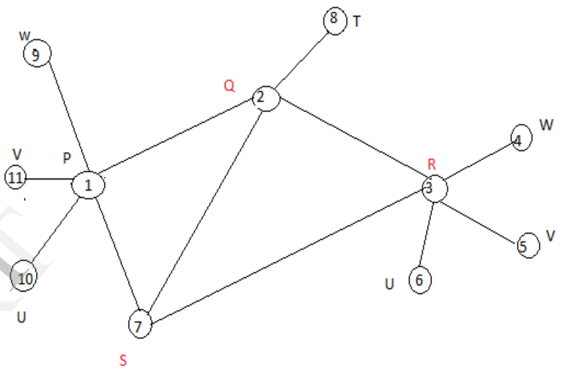


Fig. 1

IV. ALGORITHM USED

Through this algorithm we perform grouping of nodes, and then carry out appropriate alteration of neighbors tagsof every nodes of each group to satisfy the ℓ -private-tag-miscellany requirement. Nodes with similar sidewise locality information are grouped together so that we can alter few tags as possible and noisy nodes are added if required. The two algorithms that we present modify tags add edges and add noisy nodes. In starting we use, False Node Algorithm (FNA) that sort nodes by degree and compare sidewise locality information of nodes with similar degree. In this we make an assumption that nodes with similar degree are more likely to have similar sidewise locality information. So, we expect to reduce computation by not repeating the calculation of similarity for each and every pair of nodes.

We then propose a second algorithm,Extended False Node Algorithm (EFNA).

A. *FNA (False Node Algorithm)*

Algorithm starts out by sorting nodes (N) by degree. Nodes with theequal degree, which belong to the same segment, are grouped by the similarityof their sidewise locality tags. For two nodes, n1 with sidewise locality tag set(NL_1), and n2 with sidewise locality tag set (NL_2), we calculate sidewise localitytag similarity (Ls) as follows:

Ls (n1, n2) = | NL1 ∩ NL2 | ÷ |NL1 ∪ NL2| (1)

False Node Algorithm (FNA)
Input: graph G (N, E, L_o, L) parameter g, M_s= maximum similarity;
Result: Modified Graph G’

```
Firstly, sort nodes by degree
for each segment in graph do
 if ($L_o$ >g && $L_o$==g) then
Calculate similarity among nodes;
G ←v1, v2 with $M_s$;
While   g >|G| do
GroupG ← next n;
Neighbors of G are modified;
While 0 >$N_{left}$ do
  Ifℓ ≤ |$N_{left}$ |then
  Similarities are computed pairwise;
Group G  ← v1, v2with$M_s$;
While ℓ> |G| do
  GroupG ← next v;
  All the neighbors of G are modified;
Elseif ℓ>| $N_{left}$ |then
 For each n ϵ E |$N_{left}$ |do
  Similarity (n,$G_0$);
  $G_{maximum\ \ similarity}$ ←V;
  $G_{maximum\ \ similarity}$ Neighbors are modified;
 Return G’ (N’, E’, L’);
```

B. EFNA (Extended False Node Algorithm)

For two nodes, n1 with sidewise locality tag set (NL_1), and n2 with sidewise localitytag set (NL_2), we calculate sidewise locality tag similarity (L_S) as follows:

L_S(n1, n2) = |NL1∩ $NL2$ | ÷ |NL1 U NL2| (2)

Input: graph G (N, E, L,L_0), parameter g, x;
Result: Modified Graph G’

```
 while 0<$N_{LEFT}$ do
  Ifℓ≤ $N_{LEFT}$
Calculate similarity among nodes;
 Group G  ← v1, v2with$M_s$;
 All the neighbors of G are modified;
 Whileℓ>|G| do
 Dissimilarity ($N_{LEFT}$ , G);
 Group G  ← v with$M_S$;
 Without adding noisy nodes neighbors of graph G are modified;
  Elseif ℓ> |$N_{LEFT}$| then
 for each n ∈ $N_{LEFT}$ do
 Similarity (n, $G_0$);
 $G_{maximum\ \ similarity}$ ←n;
 Without adding noisy nodes neighbors of graph $G_{maximum\ \ similarity}$ are modified;
 Expected noisy nodes are added if required;
Return G’ (N’, E’, L’);
```

V. COMPARISION

A. FNA(False Node Algorithm)

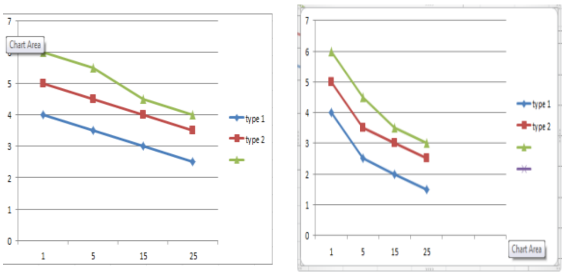
- False Node Algorithm (FNA) sort nodes by degree and compare sidewise locality information of nodes with similar degree. Nodes with similar degree are more likely to have similar sidewise locality information .thus we expect to reduce computation by avoiding recounting similarity for every pair of nodes.
- Firstly algorithm starts by sorting vertices with the same degree, which belong to the same segment, are grouped by the similarity of their sidewise locality tags.
- Once the similarity is computed between pairs of nodes in the same segment, two nodes (m, n) with the maximal similarity value are grouped together. Other nodes are then merged to this group according to their similarities with either m or n, till this group obtains nodes with different private tags. After one group is finalized, we calculate the number of nodes left in the segment to check whether there are enough nodes for forming another group.
- Sidewise locality tags are modified right after every grouping operation, so the tags of nodes can be accordingly updated immediately for the next group in the next grouping operation. This modification process ensures that all nodes in a group have same sidewise locality information.

B. EFNA (Extended False Node Algorithm)

- Extended False Node Algorithm (GINN), computes dissimilarity based on both degree and tags.
- No priority is given to degree and there is no sorting step.
- During group formation, all nodes that have not yet been grouped are taken into consideration, in clustering-like fashion.
- Initially two nodes are selected with the minimum dissimilarity and they are grouped together. Their neighbor tags are modified to be the same immediately so that nodes in one group always have the same neighbor tags.
- Nodes having the minimum dissimilarity with any node in the group are clustered into the group till the group has l nodes with different private tags. After that, the algorithm proceeds to creating the next group. If there are fewer than nodes left after the last group's formation, these remainder nodes are clustered into existing groups according to the similarities between nodes and groups

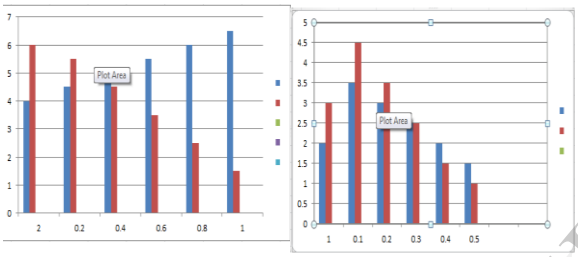
On the basis of degree, noisy nodes and noisy edges we try to show difference on basis of facts account we conserve from original graph. Charts show degree comparison b/w FNA and

EFNA. As, from graph we conclude that degree distribution in EFNA is better than FNA.
Y-axis- shows frequency of nodes, X-axis- shows degree



(False Node Algorithm) (Extended False Node Algorithm)
Fig.2

In Fig 3 we show two graphs one for noisy nodes and another for noisy edges that are added in the graph to maintain the privacy. The amount of noisy nodes and edges gives the idea of imminent ruination of facts account after alteration. Comparing the two algorithm, we conclude that algorithm EFNA always brings the fewest noisy nodes and edges, and thus the least imminent to the graphs structure.



(Noisy Nodes)(Noisy Edges)
Fig.3

VI. CONCLUSION

In this, paper we propose algorithms FNA and EFNA that shows how we can preserve our data on the social networking site from adversary. In this we make some label as private and non-private and then we show comparison in this two algorithm. The graphs are turned into algorithm and we conclude that EFNA shows better result than FNA.

REFERENCES

[1] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. `-diversity: privacy beyond k-anonymity. In ICDE, 2006.

[2] Personal privacy vs. population privacy: learning to attack anonymization. In SIGKDD, 2011.

[3] L. Backstrom, C. Dwork, and J. M. Kleinberg. Wherefore art thou R3579X?: anonymized social networks, hidden patterns, and structural steganography. Com- mun. ACM, 54(12), 2011.

[4] L. Sweeney. K-anonymity: a model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(5), 2002

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

[6] X. Ying and X. Wu. Randomizing social networks: a spectrum perserving approach. In SDM, 2008.

[7] J. Cheng, A. W.-C. Fu, and J. Liu. K-isomorphism: privacy-preserving network publication against structural attacks. In SIGMOD, 2010

[8] A. G. Francesco Bonchi and T. Tassa. Identity obfuscation in graphs through the information theoretic lens. In ICDE, 2011.

[9] E. Zheleva and L. Getoor. Preserving the privacy of sensitive relationships in graph data. In PinKDD, 2007.

[10] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. S. and. Class-based graph anonymization for social network data. PVLDB, 2(1), 2009.

[11] B. Zhou and J. Pei. The k-anonymity and `-diversity approaches for privacy preser- vation in social networks against neighborhood attacks. Knowledge and Informa- tion Systems, 28(1), 2010

[12] M. Yuan, L. Chen, and P. S. Yu. Personalized privacy protection in social networks. PVLDB, 4(2), 2010.

[13] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. `-diversity: privacy beyond k-anonymity. In ICDE, 2006.