Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 7, July 2014, pg.706 – 715

RESEARCH ARTICLE

Sensitive Label Privacy Protection on Social Network Data

B.Bhasker[1], M.Vazralu[2], G.Anitha[3]

[1,2,3] C.S.E and JNTUH

[1] Bhasker.b90@gmail.com, [2] Vazram4u@gmail.com, [3] Ganitha29685@gmail.com

[1]M. Tech, IV semester, Department of CSE, Malla Reddy College of Engineering and Technology, Hyderabad

[2]Associate Professor, Department of CSE, Malla Reddy College of Engineering and Technology, Hyderabad

[3]Associate Professor, Department of CSE, Malla Reddy College of Engineering and Technology, Hyderabad

Abstract— The advent of the Web 2.0 has caused social profiling and is a growing concern for internet privacy.[1] Web 2.0 is the system that facilitates participatory information sharing and collaboration on the Internet, in social networking media websites like Facebook and MySpace.[1] These social networking sites have seen a boom in their popularity starting from the late 2000s. Through these websites many people are giving their personal information out on the internet. These social networks keep track of all interactions used on their sites and save them for later use.[2] Issues include cyberstalking, location disclosure, social profiling, 3rd party personal information disclosure, and government use of social network websites in investigations without the safeguard of a search warrant This paper is motivated by the recognition of the need for a finer grain and more personalized privacy in data publication of social networks. We propose a privacy protection scheme that not only pre-vents the disclosure of identity of users but also the disclosure of selected features in users' pro les. An individual user can select which features of her pro le she wishes to conceal. The social networks are modelled as graphs in which users are nodes and features are labels. Labels are denoted either as sensitive or as non-sensitive. We treat node labels both as background knowledge an adversary may possess, and as sensitive information that has to be protected. We present privacy protection algorithms that allow for graph data to be published in a form such that an adversary who possesses information about a node's neighborhood cannot safely infer its identity and its sensitive labels. To this aim, the algorithms transform the original graph into a graph in which nodes are sufficiently indistinguishable. The algorithms are designed to do so while losing as little information and while preserving as much utility as possible. We evaluate empirically the extent to which the algorithms preserve the original graph's structure and properties. We show that our solution is effective, efficient and scalable while offering stronger privacy guarantees than those in previous research

1 Introduction

The publication of social network data entails a privacy threat for their users. Sensitive information about users of the social networks should be protected. The challenge is to devise methods to publish social network data in a form that affords utility without compromising privacy. Previous research has pro-posed various privacy models with the corresponding protection mechanisms that prevent both inadvertent private information leakage and attacks by malicious adversaries. These early privacy models are mostly concerned with identity and link disclosure. The social networks are modelled as graphs in which users are nodes and social connections are edges. The threat definitions and protection mechanisms leverage structural properties of the graph. This paper is motivated by the

recognition of the need for a ner grain and more personalized privacy.

Users entrust social networks such as Facebook and LinkedIn with a wealth of personal information such as their age, address, current location or political orientation. We refer to these details and messages as features in the user's pro le. We propose a privacy protection scheme that not only prevents the disclosure of identity of users but also the disclosure of selected features in users' pro les. An individual user can select which features of her pro le she wishes to conceal.

The social networks are modeled as graphs in which users are nodes and features are labels[1]. Labels are denoted either as sensitive or as non-sensitive. Figure 1 is a labeled graph representing a small subset of such a social network. Each node in the graph represents a user, and the edge between two nodes represents the fact that the two persons are friends. Labels annotated to the nodes show the locations of users. Each letter represents a city name as a label for each node. Some individuals do not mind their residence being known by the others, but some do, for various reasons. In such case, the privacy of their labels should be protected either sensitive (labels are in red

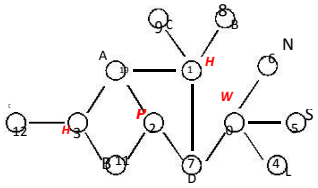


Fig. 1. Example of the labeled graph representing a social network

The privacy issue arises from the disclosure of sensitive labels. One might suggest that such labels should be simply deleted. Still, such a solution would present an incomplete view of the network and may hide interesting statisti-cal information that does not threaten privacy. A more sophisticated approach consists in releasing information about sensitive labels, while ensuring that the identities of users are protected from privacy threats. We consider such threats as neighborhood attack, in which an adversary nds out sensitive information based on prior knowledge of the number of neighbors of a target node and the labels of these neighbors. In the example, if an adversary knows that a user has three friends and that these friends are in A (Alexandria), B (Berlin) and C (Copenhagen), respectively, then she can infer that the user is in H (Helsinki).

We present privacy protection algorithms that allow for graph data to be published in a form such that an adversary cannot safely infer the identity and sensitive labels of users. We consider the case in which the adversary possesses both structural knowledge and label information.

The algorithms that we propose transform the original graph into a graph in which any node with a sensitive label is indistinguishable from at least ` 1 other nodes. The probability to infer that any node has a certain sensitive label (we call such nodes sensitive nodes) is no larger than 1=`. For this purpose we design `-diversity-like model, where we treat node labels as both part of an adversary's background knowledge and as sensitive information that has to be protected.

The algorithms are designed to provide privacy protection while losing as little information and while preserving as much utility as possible. In view of the tradeo between data privacy and utility [16], we evaluate empirically the extent to which the algorithms preserve the original graph's structure and properties such as density, degree distribution and clustering coefficient. We show that our solution is effective, efficient and scalable while offering stronger privacy guarantees than those in previous research, and that our algorithms scale well as data size grows. Privacy is one of the major concerns when publishing or sharing social network data for social science research and business analysis. Recently, researchers have developed privacy models similar to k-anonymity to prevent node reidentification through structure information. However, even when these privacy models are enforced, an attacker may still be able to infer one's private information if a group of nodes largely share the same sensitive labels (i.e., attributes). In other words, the label-node relationship is not well protected by pure structure anonymization methods. Furthermore, existing approaches, which rely on edge editing or node clustering, may significantly alter key graph properties. In this paper, we define a k-degree-l-diversity anonymity model that considers the protection of structural information as well as sensitive labels of individuals. We further propose a novel anonymization methodology based on adding noise nodes. We develop a new algorithm by adding noise nodes into the original graph with the consideration of introducing the least distortion to graph properties. Most importantly, we provide a rigorous analysis of the theoretical bounds on the number of noise nodes added and their impacts on an important graph property. We conduct extensive experiments to evaluate the effectiveness of the proposed technique.

The rest of the paper is organized as follows. Section 2 reviews previous works in the area. We de ne our problem in Section 3 and propose solutions in Section 4. Experiments and result analysis are described in Section 5. We conclude this work in Section 6.

2 Literature Review

The rst necessary anonymization technique in both the contexts of micro- and network data consists in removing identification. This nave technique has quickly been recognized as failing to protect privacy. For microdata, Sweeney et al. propose k-anonymity [17] to circumvent possible identity disclosure in naively anonymized microdata. `-diversity is proposed in [13] in order to further prevent attribute disclosure.

Similarly for network data, Backstrom et al., in [2], show that naive anonymiza-tion is insufficient as the structure of the released graph may reveal the identity of the individuals corresponding to the nodes. Hay et al. [9] emphasize this problem and quantify the risk of re-identi cation by adversaries with external informa-tion that is formalized into structural queries (node re nement queries, subgraph knowledge queries). Recognizing the problem, several works [5, 11, 18, 20{22, 24, 27, 8, 4, 6] propose techniques that can be applied to the naive anonymized graph, further modifying the graph in order to provide certain privacy guarantee. Some works are based on graph models other than simple graph [12, 7, 10, 3].

To our knowledge, Zhou and Pei [25, 26] and Yuan et al. [23] were the rst to consider modeling social networks as labeled graphs, similarly to what we con-sider in this paper. To prevent re-identi cation attacks by adversaries with imme-diate neighborhood structural knowledge, Zhou and Pei [25] propose a method that groups nodes and anonymizes the neighborhoods of nodes in the same group by generalizing node labels and adding edges. They enforce a k-anonymity pri-vacy constraint on the graph, each node of which is guaranteed to have the same immediate neighborhood structure with other k 1 nodes. In [26], they improve the privacy guarantee provided by k-anonymity with the idea of `-diversity, to protect labels on nodes as well. Yuan et al. [23] try to be more practical by considering users' different privacy concerns. They divide privacy requirements into three levels, and suggest methods to generalize labels and modify structure corresponding to every privacy demand. Nevertheless, neither Zhou and Pei, nor Yuan et al. consider labels as a part of the background knowledge. However, in case adversaries hold label information, the methods of [25, 26, 23] cannot achieve the same privacy guarantee. Moreover, as with the context of microdata, a graph that satisfies a k-anonymity privacy guarantee may still leak sensitive information regarding its labels [13].

Existing System

The current trend in the Social Network it not giving the privacy about user profile views. The method of data sharing or (Posting) has taking more time and not under the certain condition of displaying sensitive and non-sensitive data.

Problems on existing system:

1. There is no way to publish the Non sensitive data to all in social Network.
2. It's not providing privacy about user profiles.
3. Some mechanisms that prevent both inadvertent private information leakage and attacks by malicious adversaries.

Proposed System

Here, we extend the existing definitions of modules and we introduced the sensitive or non-sensitive label concept in our project. We overcome the existing system disadvantages in our project.

Advantages:

1. We can publish the Non sensitive data to every-one in social Network.
2. It's providing privacy for the user profiles so that unwanted persons not able to view your profiles.
3. We can post sensitive data to particular peoples and same way we can post non-sensitive data to everyone like ads or job posts

IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Main Modules:-

1. User Module:

In this module, Users are having authentication and security to access the detail which is presented in the ontology system. Before accessing or searching the details user should have the account in that otherwise they should register first.

2. Information Loss:

We aim to keep information loss low. In- formation loss in this case contains both structure information loss and label information loss. There are some non sensitive data's are Loss due to Privacy making so we can't send out full information to the public.

3. Sensitive Label Privacy Protection:

There are who post the image to the online social network if allow the people for showing the image it will display to his requesters it make as the sensitive to that user. Thesis is very useful to make sensitive data for the public .

System Configuration:-

H/W System Configuration:-

Processor - Pentium –III

Speed - 1.1 Ghz

RAM - 256 MB(min)

Hard Disk - 20 GB

Floppy Drive - 1.44 MB

Key Board - Standard Windows Keyboard

Mouse - Two or Three Button Mouse

Monitor - SVGA

S/W System Configuration:-

- Operating System :Windows95/98/2000/XP
- Application Server : Tomcat5.0/6.X
- Front End : HTML, Java, Jsp
- Scripts : JavaScript.
- Server side Script : Java Server Pages.
- Database : Mysql 5.0
- Database Connectivity : JDBC.

3. **Problem Definition**

We model a network as G(V; E; L^S; L;), where V is a set of nodes, E is s set of edges, L^S is a set of sensitive labels, and L is a set of non-sensitive labels. maps nodes to their labels, : V ! L^S [L. Then we propose a privacy model, `-sensitive-label-diversity; in this model, we treat node labels both as part of an adversary's background knowledge, and as sensitive information that has to be protected. These concepts are clarified by the following definitions:

Definition 1. The neighborhood information of node v comprises the degree of v and the labels of v's neighbors.

Definition 2. (`-sensitive-label-diversity) For each node v that associates with a sensitive Neighborhood with the same labels.

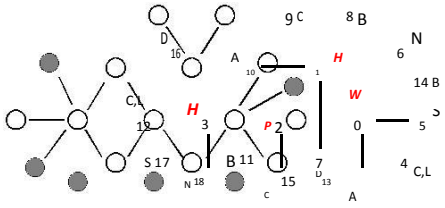


Fig. 2. Privacy-attaining network example

In Example 1, nodes 0, 1, 2, and 3 have sensitive labels. The neighborhood information of node 0, includes its degree, which is 4, and the labels on nodes 4, 5, 6, and 7, which are L, S, N, and D, respectively. For node 2, the neigh-borhood information includes degree 3 and the labels on nodes 7, 10, and 11, which are D, A, and B. The graph in Figure 2 satis es 2-sensitive-label-diversity; that is because, in this graph, nodes 0 and 3 are indistinguishable, having six neighbors with label A, B, fC,Lg, D, S, N separately; likewise, nodes 1 and 2 are indistinguishable, as they both have four neighbors with labels A, B, C, D separately.

4. Algorithm

The main objective of the algorithms that we propose is to make suitable group-ing of nodes, and appropriate modi cation of neighbors' labels of nodes of each group to satisfy the l-sensitive-label-diversity requirement. We want to group nodes with as similar neighborhood information as possible so that we can change as few labels as possible and add as few noisy nodes as possible. We propose an algorithm, Global-similarity-based Indirect Noise Node (GINN), that does not attempt to heuristically prune the similarity computation as the other two algo-rithms, Direct Noisy Node Algorithm (DNN) and Indirect Noisy Node Algorithm (INN) do. Algorithm DNN and INN, which we devise rst, sort nodes by degree and compare neighborhood information of nodes with similar degree. Details about algorithm DNN and INN please refer to [15].

4.1 Algorithm GINN

The algorithm starts out with group formation, during which all nodes that have not yet been grouped are taken into consideration, in clustering-like fashion. In the rst run, two nodes with the maximum similarity of their neighborhood labels are grouped together. Their neighbor labels are modi ed to be the same immediately so that nodes in one group always have the same neighbor labels. For two nodes, v_1 with neighborhood label set (LS_{V1}), and v_2 with neighborhood label set (LS_{V2}),
we calculate neighborhood label similarity (NLS) as follows:

$$NLS(v_1 ; v_2) = \frac{j\ LS_{v_1} \setminus LS_{v_2}\ j}{j\ LS_{v_1} [LS_{v_2}\ j} \qquad (1)$$

Larger value indicates larger similarity of the two neighborhoods.

Then nodes having the maximum similarity with any node in the group are clustered into the group till the group has ` nodes with di erent sensitive labels. Thereafter, the algorithm proceeds to create the next group. If fewer than ` nodes are left after the last group's formation, these remainder nodes are clustered into existing groups according to the similarities between nodes and groups.

After having formed these groups, we need to ensure that each group's mem-bers are indistinguishable in terms of neighborhood information. Thus, neigh-borhood labels are modifieed after every grouping operation, so that labels of nodes can be accordingly updated immediately for the next grouping opera-tion. This modification process ensures that all nodes in a group have the same neighborhood information. The objective is achieved by a series of modification operations. To modify graph with as low information loss as possible, we devise three modification operations: label union, edge insertion and noise node addition. Label union and edge insertion among nearby nodes are preferred to node addition, as they incur less alteration to the overall graph structure.

Edge insertion is to complement for both a missing label and insufficicient degree value. A node is linked to an existing nearby (two-hop away) node with that label. Label union adds the missing label values by creating super-values shared among labels of nodes. The labels of two or more nodes coalesce their values to a single super-label value, being the union of their values. This approach maintains data integrity, in the sense that the true label of node is included among the values of its label super-value. After such edge insertion and label union operations, if there are nodes in a group still having different neighborhood information, noise nodes with non-sensitive labels are added into the graph so as to render the nodes in group indistinguishable in terms of their neighbors' labels. We consider the uni cation of two nodes' neighborhood labels as an example. One node may need a noisy node to be added as its immediate neighbor since it does not have a neighbor with certain label that the other node has; such a label on the other node may not be modi able, as its is already connected to another sensitive node, which prevents the re-modification on existing modified groups.

Algorithm 1: Global-Similarity-based Indirect Noisy Node Algorithm

Input: graph G(V; E; L; L^s), parameter l;
Result: Modi ed Graph G^0

```
while V_left > 0 do
    if |V_left| < l then
        compute pairwise node similarities;
        group G{v1; v2} with Max_similarity;
        Modify neighbors of G;
        while |G| < l do
            dissimilarity(V_left; G);
            group G{v} with Max_similarity;
            Modify neighbors of G without actually adding noisy nodes ;
    else if |V_left| < l then
        for each v ∈ V_left do
            similarity(v; Gs);
            G{v, Max similarity}
        Modify neighbors of G_{Max similarity} without actually adding noisy nodes;
Add expected noisy nodes;
Return G^0(V^0; E^0; L^0);
```

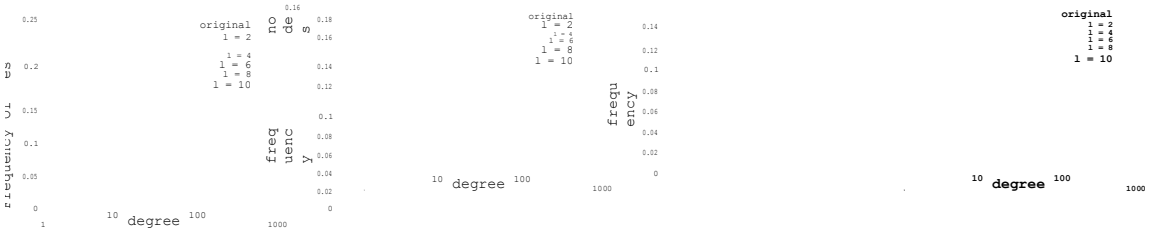
In this algorithm, noise node addition operation that is expected to make the nodes inside each group satisfy `-sensitive-label-diversity are recorded, but not performed right away. Only after all the preliminary grouping operations are performed, the algorithm proceeds to process the expected node addition operation at the nal step. Then, if two nodes are expected to have the same labels of neighbors and are within two hops (having common neighbors), only one node is added. In other words, we merge some noisy nodes with the same label, thus resulting in fewer noisy nodes.

5. Experimental Evaluation

We evaluate our approaches using both synthetic and real data sets. All of the approaches have been implemented in Python. The experiments are conducted on an Intel core, 2Quad CPU, 2:83GHz machine with 4GB of main memory running Windows 7 Operating System. We use three data sets. The rst data set [1] is a network of hyperlinks between weblogs on US politics. The second data set that we use is generated from the Facebook dataset proposed in [14]. The third data set that we use is a family of synthetic graphs with varying number of nodes. The rst and second datasets are used for the evaluation of effectiveness (utility and information loss). The third data set is used to measure runtime and scalability (running time). (Please refer to [15] for more information.)

5.1 Data Utility

We compare the data utilities we preserve from the original graphs, in view of measurements on degree distribution, label distribution, degree centrality [19], clustering coefficient, average path length, graph density, and radius. We show the number of the noisy nodes and edges needed for each approach.



(a) DNN (b) INN (c) GINN

Fig. 3. Facebook Graph Degree Distribution

Figure 3 shows the degree distribution of the Facebook graph both before and after modification. Each sub gure in Figure 3 shows degree distributions of graphs modified by one algorithm. We can see that the degree distributions of the modified graphs resemble the original ones well, especially when l is small.

To sum up, these measurements (for other results please refer to [15]) show that the graph structure properties

are preserved to a large extent. The strong resemblance of the label distributions in most cases indicates that the label infor-mation, another aspect of graph information, is well maintained. They suggest as well that algorithm GINN does preserve graph properties better than the other two while these three algorithms achieve the same privacy constraint.

5.2 Information Loss

In view of utility of released data, we aim to keep information loss low. In-formation loss in this case contains both structure information loss and label v's original labels and l_v^0 the set of labels in the modified graph. Thus, for the modified graph including n noisy nodes, and m noisy edges, information loss is defined as

information loss. We measure the loss in the following $_0$ way: for any node v 2 V , label dissimilarity is de ned as: $D(l_v ; l_v^U) = 1 \quad \frac{|l_v \setminus l_v|}{|l_v \cup l_v|}$, where l_v is the set of

$$IL = !_1 n + !_2 m + (1 \ !_1 \ !_2) \ D(l_v; l_v^0) \qquad (2)$$

where $!_1$,$!_2$ and $1 \ !_1 \ !_2$ are weights for each part of the information loss. Figure 4 shows the measurements of information loss on the synthetic data set using each algorithm. Algorithm GINN introduces the least information loss.

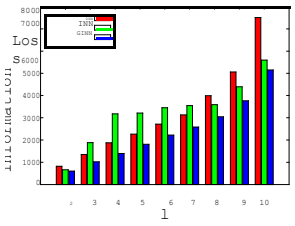


Fig. 4. Information Loss

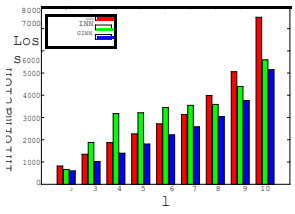


Fig. 5. Running Time

5.3 Algorithm Scalability

We measure the running time of the methods for a series of synthetic graphs with varying number of nodes in our third dataset. Figure 5 presents the running time of each algorithm as the number of nodes increases. Algorithm DNN is faster than the other two algorithms, showing good scalability at the cost of large noisy nodes added. Algorithm GINN can also be adopted for quite large graphs as fol-lows: We separate the nodes to two different categories, with or without sensitive labels. Such smaller granularity reduces the number of nodes the anonymization method needs to process, and thus improves the overall efficiency.

6. **Conclusions**

In this paper we have investigated the protection of private label information in social network data publication. We consider graphs with rich label information, which are categorized to be either sensitive or non-sensitive. We assume that adversaries possess prior knowledge about a node's degree and the labels of its neighbors, and can use that to infer the sensitive labels of targets. We suggested a model for attaining privacy while publishing the data, in which node labels are both part of adversaries' background knowledge and sensitive information that has to be protected. We accompany our model with algorithms that transform a network graph before publication, so as to limit adversaries' confidence about sensitive label data. Our experiments on both real and synthetic data sets con-rm the effectiveness, efficiency and scalability of our approach in maintaining critical graph properties while providing a comprehensible privacy guarantee.

References

1. Dwyer, C., Hiltz, S. & Passerini, K. (2007). Trust and Privacy Concern within Social Networking Sites: A Comparison of Facebook and MySpace. Americas Conference on Information Systems. Retrieved from http://google.com/scholar?q=cache:qLCk18d_wZwJ:scholar.google.com/+facebook+privacy&hl=en&as_sdt=2000
2. Tracy Mitrano. (2006, November, December). A Wider World: Youth, Privacy, and Social Networking Technologies. Retrieved fromhttp://www.educause.edu/EDUCAUSE+Review/EDUCAUSEReviewMagazineVolume41/AWiderWorldYouthPrivacyandSoci/158095
3. Retrieved from: http://abcnews.go.com/Technology/smartphone-apps-tracking-keeping-tabs-past-lovers-people/story?id=13022144
4. EPIC – In re Facebook. (n.d.). EPIC – Electronic Privacy Information Center. Retrieved January 25, 2011/
5. Danesh Irani, Steve Webb, Calton Pu, Kang Li, "Modeling Unintended Personal-Information Leakage from Multiple Online Social Networks," IEEE Internet Computing, May/June 2011. Retrieved from http://www.computer.org/csdl/mags/ic/2011/03/mic2011030013-abs.html
6. Balachander Krishnamurthy, Konstantin Naryshkin, Craig Wills, "Privacy leakage vs. Protection measures: the growing disconnect," Web 2.0 Security and Privacy Workshop, May 2011. Retrieved from http://www.research.att.com/~bala/papers/w2sp11.pdf
7. Balachander Krishnamurthy and Craig Wills, "On the Leakage of Personally Identifiable Information Via Online Social Networks," Proceedings of ACM SIGCOMM Workshop on Online Social Networks, August 2009. Retrieved from http://www.research.att.com/~bala/papers/wosn09.pdf
8. Facebook's Privacy Policy. (2010). Retrieved from http://www.facebook.com/policy.php
9. Chai S, Bagchi-Sen S, Morrell C, Rao H, Upadhyaya S. Internet and online information privacy: An exploratory study of preteens and early teens. IEEE Transactions On Professional Communication [serial online]. June 2009;52(2):167–182. Available from: PsycINFO, Ipswich, MA. Accessed February 6, 2012.
10. Moscardelli D, Divine R. Adolescents' Concern for Privacy When Using the Internet: An Empirical Analysis of Predictors and Relationships with Privacy-Protecting Behaviors. Family And Consumer Sciences Research Journal [serial online]. March 2007;35(3):232–252. Available from: PsycINFO, Ipswich, MA. Accessed February 6, 2012.
11. Richard Lardner. (2010, March 16). Your new Facebook 'friend' may be the FBI. Retrieved from http://www.msnbc.msn.com/id/35890739/ns/technology_and_science-security/
12. Harkins, Gina. (2011, March 02). Cops patrol social networking sites for gang activity. Retrieved from http://news.medill.northwestern.edu/chicago/news.aspx?id=181375
13. Taghi, Hasti. (2011, February 10). Police Use Facebook To Track Suspect. Retrieved from http://www.click2houston.com/news/26825687/detail.html
14. Halverstadt, Lisa. (2009, March 12). Surprise police use MySpace to locate teen graffiti suspect. Retrieved from http://www.azcentral.com/news/articles/2009/03/12/20090312gl-nwvmyspace0313.html
15. http://cb.hbsp.harvard.edu/cb/web/he/product_view.seam?R=808128-PDF-ENG&T=EDC&C=PURCHASED_MATERIALS&CD=16117304&CS=90ae2ef93335be224703aca3962ad383

Authors Biography:

First Author :B.Bhasker

M.Tech in CSE from JNTUH,
Malla Reddy College of Engineering and Technology,
Hyderabad

Second Author M.Vazralu

M.Tech in CSE from JNTUH,
Assoc Prof,Dept of CS&E,
Malla Reddy College of Engineering and Technology,
Hyderabad

Third Author G.Anitha

M.Tech in CSE from JNTUH,
Assoc Prof,Dept of CS&E,
Malla Reddy College of Engineering and Technology,
Hyderabad

