International Journal of Engineering & Technology, 5 (x) (2017) xxx-xxx

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: xxx
Research paper, Short communication, Review, Technical paper

Privacy Preservation in Sequential Published Social Networks Against Mutual Friend Attack

Jyothi Vadisala[1*] and Valli Kumari Vatsavayi[2]

[1]*Dept. of CS & SE, AUCE(A), Andhra University, Visakhaptnam, India*
[2]*Professor, Dept. of CS & SE, AUCE(A), Andhra University, Visakhaptnam, India*
[*]*Corresponding author E-mail: jyothi.vadisala@gmail.com*

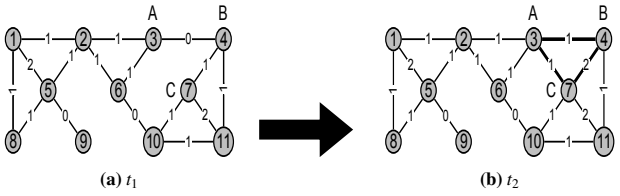
Abstract

In recent years the social networks are widely used the way of connecting people, interact with each other and share the information. The social network data is rich in content and the data are published for third party users such as researchers. The social interaction between individuals changes rapidly as time changes so there is a need of privacy preserving in dynamic networks. An adversary can acquire some local knowledge about individuals in the network and can easily breach the privacy of a few victims. This paper mainly focus on preserving privacy in sequential published network data where the adversary has some knowledge about the number of mutual friends of the target victims over a time period. The k^w-Number of Mutual Friend Anonymization model is proposed to anonymize each sequential published network. In this privacy model, k indicates the privacy level and w is the time interval taken by the adversary to acquire the knowledge of the victim. By this approach the adversary can not identify the victim by acquiring the knowledge of each sequential published data. The performance evaluation shows that the proposed approach can preserve many characteristics of the dynamic social networks.

Keywords: *social network, privacy, dynamic, anonymize, mutual friend.*

1. Introduction

Social Network sites like Facebook, Linkedin and Twitter are mostly used for connecting, interacting, communicating with each other and as well as share their information on the web. Due to the enormous growth of social network data from many applications and services, it is published for various research purposes. This leads to a lot of privacy issues as it can leak highly sensitive data of many individuals and groups. When the network data have to be released it should be anonymized and published for various purposes. Different anonymization techniques are used for preserving the social network data publication on various background knowledge attacks[1][2]. Liu and Terzi in [5] proposed k-degree anonymity based on the degree knowledge to defend the vertex identification. Zhou and Pei [6] proposed k-neighborhood anonymity based on the knowledge of victims one-neighbor graph. Lei Zou et al. in [7] proposed k-automorphism based on the knowledge of subgraph of an arbitrary size. Hay et al.[4] proposed clustering technique to group vertices and edges to protect the vertex identification. Tai and Yu in [8] proposed the friendship attack model, where an adversary knows the vertex degree pair of two individuals and their friendship relation. In this, they solved friendship attack problem based on the degree of two concerned nodes. Chongjing Sun et al. [15], proposed a relationship model taking the number of mutual friends of the connected nodes into account and proposed an algorithm in which they ensure that at least $k-1$ other friend pairs that share same number of mutual friends by preserving the original vertex set. These works mainly focus on issues in the static network data. Network data changes dynamically with time and is valuable for research purpose. As there is a high demand for the dynamic social network analysis, privacy issues of each sequential release of dynamic social network data become important. Bhagat et al. [9], demonstrated the procedure on how to protect the association of labels between vertices which are directly connected but the identity protection of these vertices are not addressed. Medforth and Wang extended [10] the degree attack model for dynamic networks and has mentioned the privacy breach in sequential releases but they did not build models for privacy preservation. Suppose a network is published sequentially and anonymized on each release still there is a possibility to intrude privacy.



(a) t_1 **(b)** t_2

Figure 1: Sequential published dynamic network.

Example 1. *Motivation: Figure 1 shows a sequentially published dynamic friendship relation naive anonymized network in which each vertex represents a user. The value on the edge connecting two end vertices represents the number of mutual friends they have. Suppose that user's friends count is known to attacker (degree attack), the attacker cannot uniquely re-identify the user in the released network*

at time t_1 and t_2. Now, suppose user knows the common friends count of the two end vertices which is known as Mutual Friend Attack then the adversary has difficulty to re-identify anyone from sequentially published dynamic social network. From the above sequentially published data there are two problems where the adversary can trace the user. In the first case, if an adversary knows that both Bob and Carl are friends and recently Alice and Carl have become friends. An adversary cannot uniquely identify the victims from these sequentially released networks. From the static networks the adversary cannot identify the edges that are newly added but, he can easily find that (v_3, v_7) is the newly added edge by comparing two sequentially released networks. The new edge represents both Alice and Carl's friendship relation. With this background knowledge the adversary can also confirm that one more mutual friend for Bob and Carl is added i.e Alice. An adversary cannot identify the edge which got changed in the published static networks. Instead by observing the combination of two releases, the problem will be easier because (v_4, v_7) is the only edge changing from 1 to 2 and Alice i.e v_3 is the new common friend for Bob and Carl. Hence an adversary can reveal the relationships of the individuals from sequentially published network data.

Example 1 illustrates that anonymization of the current network should be based on the present information as well as the previous released network data. The existing k-NMF anonymization algorithm cannot ensure the privacy in dynamic releases. To protect against such attack a model called dynamic k^w-NMF is proposed with the probability of an edge disclosure is limited to $1/k$ where the adversary can monitor a victim within a continuous time period w. The proposed model anonymizes the current network based on the previous $w-1$ releases and minimizes the graph alternations. The results show that the proposed method can preserve most of the characteristics of the original graph with limited information distortion.

2. Problem Formulation

Generally a Social Network is modeled as a simple undirected graph $G(V,E)$, where V is a set of vertices representing individuals, and $E \subseteq V \times V$ is the set of edges representing the relationship of individuals. *"We formulate dynamic network as time-stamped graph, as time varies, new individuals will be participating in dynamic network and the relationship between individuals also changes"*. Let t be time instance and $G^t(V^t, E^t)$ is denoted as a dynamic network at any given time t, where V^t is the set of vertices at t instance representing the users, E^t is the set of edges at time instance t representing the relationship between users. G^t is a dynamic network and the published or released graph of G^t will be denoted as $\bar{G}^t$ in the remainder of this paper.

Definition 1. *(Number of Mutual Friends of an Edge): The mutual friends count of two connected vertices v_1 and v_2 of an edge e in a graph $\bar{G}^t(V^t, E^t)$ at time t, i.e $v_1, v_2 \in V^t$, $e \in E^t$ and $e = (v_1, v_2)$, the mutual friends count of the edge e is the number of common neighbors of both v_1 and v_2 i.e., $nmf(e) = Neigh(v_1) \cap Neigh(v_2)$*

Definition 2. *(NMF Sequence): Let γ^t denote the number of mutual friends at time t instance then the number of mutual friend sequence for $\bar{G}^t$,where entries are sorted based on decreasing order i.e. $\gamma_1 \geq \gamma_2 \geq \ldots \gamma_n$. Let ℓ^t represent the edge list corresponding to γ^t i.e. γ^t_i is the mutual friends count of the edge ℓ^t_i at time instance t.*

For example, in Figure 1a, $\gamma^{t_1} = \{2,2,1,1,1,1,1,1,1,1,1,1,1\}$ and $\ell^{t_1} = \{(v_1, v_5), (v_{11}, v_7), (v_2, v_1), (v_1, v_8), (v_5, v_2), (v_8, v_5), (v_2, v_3), (v_6, v_2), (v_3, v_6), (v_{10}, v_7), (v_4, v_7), (v_{11}, v_{10}), (v_4, v_{11})\}$. The mutual friend count distribution follows a power law property [13] which is similar to the power law distribution of the vertex degree [11].

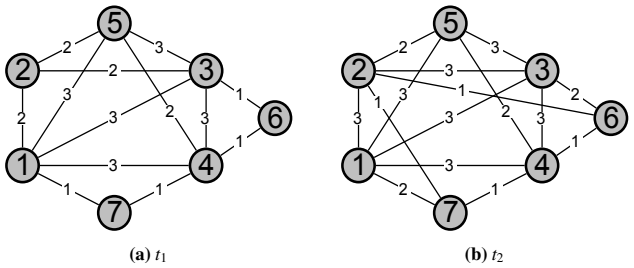


Figure 2: Example of dynamic 2^2-no. of mutual friend anonymity

Adversary Knowledge. Like previous works, the adversary has a knowledge of the mutual friends count of two connected individuals in social networking sites like Facebook, Twitter and LinkedIn. Let w be a time period where the attacker can monitor an edge e that:

1. Each release of graph during the time period w i.e., $\bar{G}^{t-w+1}, \bar{G}^{t-w+2}, \ldots, \bar{G}^t$.
2. NMF Sequence $\Gamma^w_e = \gamma^{t-w+1}_e, \gamma^{t-w+2}_e, \ldots, \gamma^t_e$ of an edge e during the time period w.

Now, we consider the base case $(w = 1)$ where the adversary has the background knowledge of a released graph $\bar{G}$ and mutual friends count of an edge γ_e, which is similar to mutual friend attack addressed in [15] and consider k-anonymous sequence for protecting the identities in which each edge in the group will have same number of mutual friends then we extend the concept to general case $(w > 1)$ for protecting edge identities in sequential releases of network data.

Definition 3. *(k-anonymous sequence): Let θ^t_γ be a sequence vector group which consists of set of edges with equal number of mutual friends γ at time t. For example, consider Fig 2a the value on each edge is the number of mutual friends of the two connected individuals. The group $\theta^{t_1}_3 = \{(1,5),(1,3),(1,4),(3,4),(3,5)\}$ is 5-anonymous group because number of mutual friends of all edges in this group is 3. A k-anonymous sequence group θ^t_γ contains at least k edges having the same number of mutual friends γ. Therefore, given a number of mutual friends γ, the probability of an edge being re-identified from θ^t_γ is limited to $1/k$.*

For dynamic releases $(w > 1)$ case, we extend the k-anonymous sequence to the sequential releases where the adversary cannot disclose the identity. To prevent from privacy breach, the graph $\bar{G}$ is anonymized based on the previous releases $\bar{G}^{t-1}, \bar{G}^{t-2}, \ldots, \bar{G}^{t-w+1}$. So the adversary no longer has the knowledge of previous releases. In other words, to protect the number of mutual friends between two vertices at any given time instance, it can be extended to the number of mutual friend sequences of over a time period w. Similar to the base case, there can be many edges having similar number of mutual friends during the period w to preserve the identity of an edge against Γ^w. Based on these considerations, first k-anonymous sequence consistent group is defined and then a privacy model k^w-NMF anonymity for dynamic networks is proposed.

Definition 4. *(k-anonymous sequence consistent group): Let Θ^w_Γ be a consistent group which contains set of edges that always share the same number of mutual friends γ^t at each time instance t during a period w.*

Example 2. *Figs (2a) and (2b) are the sequential releases at t_1 and t_2 time respectively. Consider a mutual friend sequence Γ=(3,3). The consistent group $\Theta^w_{(3,3)}$ is 5-anonymous because there is an edge subset $\{(1,5),(1,3),(1,4),(3,4),(3,5)\}$ of size 5. Similarly, $\Theta^w_{(2,3)}$ is 2-anonymous because there is an edge subset $\{(1,2),(2,3)\}$ of size 2. The k-anonymous sequence consistent group Θ^w_Γ contains at least k edges sharing same number of mutual friend sequence Γ^w to ensure that the probability of an edge being identified from Θ^w_Γ is limited to at most $1/k$.*

Algorithm 1: Pseudo code of the GS-Table Construction

Data: $G^t (1 \leq t \leq w)$, GS-Table
Result: GS-Table
foreach *edge* $e(u,v) \in G^t$ **do**
  GS-Table[e] = $neigh(u) \cap neigh(v)$
end
foreach $\Theta_{\Gamma}^{[1,t-1]} \in$ *GS-Table* **do**
  SortEdges ($\Theta_{\Gamma}^{[1,t-1]}, \Gamma^{[t,t]}$)
end
return GS-Table

Definition 5. *(Dynamic k^w-NMF Anonymity): Let's consider sequential released graphs within the time span w defined as $\bar{G}^{t-w+1}, \bar{G}^{t-w+2}, \ldots, \bar{G}^t$ dynamic k^w-NMF if for every edge $e \in E^j$, $j \in [t-w+1, t]$ and the consistent group $\Theta_{\Gamma_e}^w$, of number of mutual friend sequence γ_e is k-anonymous sequence.*

Example 3. *From Example 2, it can be noticed that $\Theta_{(3,3)}^w$ is 5-anonymous and $\Theta_{(2,3)}^w, \Theta_{(2,2)}^w$, $\Theta_{(1,2)}^w$, $\Theta_{(1,1)}^w$ and $\Theta_{(0,1)}^w$ are all 2-anonymous sequence. Therefore, the sequential release networks satisfy dynamic 2^2-NMF anonymity.*

3. Anonymizaton Approach

3.1. Overview

Dynamic k^w-NMF anonymization is more challenging than anonymization of static networks because anonymization of dynamic k^w-NMF have to consider the $w-1$ previous releases. One solution is to first consider current network alone to generate a release, then based on each previous release modify the current anonymized release to eliminate each possible privacy breach. But this method consumes more time because for every time instance, it should search for all possible attacks through the w releases. For better performance, each previous release number of mutual friend sequences of edges are gathered and summarized in a table called the GS-Table. Now, before every anonymization we incrementally update the GS-Table and generate the current network based on the GS-Table. Dynamic k^w-NMF efficiently based on $w-1$ previous releases are generated by avoiding the need to search all possible privacy breaches through the releases.

This method contains three parts, GS-Table construction, updating and the anonymization. k^w-NMF requires every edge belonging to a k-anonymous sequence consistent group. First, the GS-Table is constructed and then the edges are sorted according to their prefix of the number of mutual friend sequences and i.e, in the GS-Table, edges are sorted according to the number of mutual friends at a time span of $t-w+1, t-w+2, \ldots, t$. So the sequence groups which share a similar prefix of the number of mutual friend sequences will be adjacent and the edges will belong to most closest consistent group. When current graph information is attached small sets of edges are resorted and GS-Table is maintained incrementally. After updating GS-Table with respect to the current network at each time instance, the edges are anonymized according to their number of mutual friends ranking in GS-Table. The GS-Table quickly finds the edges with a similar number of mutual friend sequences in anonymization process. Suppose an edge e has to be anonymized first check whether e belongs to an existing k-anonymous sequence consistent group. If so, e is merged with nearest k-anonymous consistent group of GS-Table. Otherwise a new k-anonymous sequence consistent group with e and other $k-1$ edges with the same consistent group is created.

Table 1: Example of the GS-Table

(a) t_1

(1,3),(1,4), (1,5),(3,4),(3,5)	3
(1,2),(4,5) (2,3),(2,5)	2
(1,7),(4,7) (4,6),(3,6)	1

(b) t_2

(1,3),(1,4), (1,5),(3,4),(3,5)	$3 \to 3$
(1,2)	$2 \to 3$
(4,5),(2,3), (2,5)	$2 \to 2$
(1,7)	$1 \to 3$
(4,7),(4,6),(3,6)	$1 \to 1$
(2,7)	$0 \to 1$

(c) t_2

(1,3),(1,4), (1,5),(3,4),(3,5)	$3 \to 3$
(1,2),(2,3)	$2 \to 3$
(4,5),(2,5)	$2 \to 2$
(1,7)	$1 \to 3$
(4,7),(4,6), (3,6)	$1 \to 1$
(2,7),(2,6)	$0 \to 1$

(d) t_3

(3,5)	$3 \to 4$
(1,3),(1,4),(1,5) (3,4),(1,2),(2,3), (3,6),(1,7)	$3 \to 3$
(4,5),(2,5)	$2 \to 3$
(4,6),(2,6)	$1 \to 2$
(4,7,(2,7))	$1 \to 1$
(5,6)	$0 \to 3$

Algorithm 2: Pseudo code of the GS-Table Incremental Update

Data: $G^t (w \leq t)$, GS-Table
Result: GS-Table
Remove_Info($t-w$, GS-Table)
Sort_Groups($\Gamma_{[t-w+1,t-1]}$)
foreach *edge* $e(u,v) \in G^t$ **do**
  GS-Table[e] = $neigh(u) \cup neigh(v)$
end
foreach $\Theta_{\Gamma}^{[t-w+1,t-1]} \in$ *GS-Table* **do**
  SortEdges ($\Theta_{\Gamma}^{[t-w+1,t-1]}, \Gamma^{[t,t]}$)
end
return GS-Table

3.2. Construction of GS-Table

The GS-Table is constructed based on the number of mutual friend sequences of two individuals. The table consists of two columns: edge e, $e \varepsilon E$, the number of mutual friend sequence $\Gamma_e^w = \gamma_e^{t-w+1}, \gamma_e^{t-w+2}, \ldots, \gamma_e^t$. The GS-Table cannot be constructed at once because anonymizaton of dynamic graph is a continuous process. The GS-Table is constructed together with anonymization of w releases. Algorithm 1 shows the pseudo code for constructing GS-Table. Let G^1 be a given graph at time t_1 before anonymizing them GS-Table contains the edges of G^1 and the common neighbors of edges. Sort all the edges in decreasing order of their common neighbors. The GS-Table will be modified simultaneously when G^1 is anonymized. After anonymization of G^1, each edge is in a k-anonymous sequence group. Later information of G^2 is appended to the corresponding records. The edges of the same sequence group are sorted. Thus the sorting time can be reduced. After anonymization of G^2, each edge is in a k-anonymous sequence consistent group of Θ_{Γ}^2. This is continued until G^w is anonymized.

Example 4. *Consider the GS-Tables in 1 with time period $w = 2$. The edges will be in the the first column of the table and the number of mutual friend sequence will be in the second column. Table 1a shows GS-Table at time t_1 after anonymizing G^1. Suppose that the common friends of edges $(1,2), (1,7)$ and $(2,7)$ are changed to 3, 3 and 1 respectively, at time t_2 then we need to re-sort the edges in each 3-anonymous sequence consistent group $\Theta_{\Gamma}^{[1,1]}$, i.e. $\{(1,2),(4,5),(2,3),(2,5)\}$ and $\{(1,7),(4,7),(3,6),(4,6)\}$ respectively, to have the edges in decreasing order of the no. of mutual friend sequences. Table 1b shows the table after the re-sorting. After the edges are anonymized in the 2-anonymous sequence consistent group $\Theta_{\Gamma}^{[1,2]}$ are close by in the GS-Table as shown in Table 1c.*

Algorithm 3: Pseudocode for Graph Anonymization

Data: G^t, GS-Table
Result: $\bar{G}^t$, GS-Table
if $t \leq w$ **then**
  GS-TableConstruction(G^t,GS-Table)
end
else
  GS-TableIncrementalUpdate(G^t,GS-Table)
end
while $|L| \neq \emptyset$ **do**
  $GP = \{x | x \in G^t$ and $\Gamma^{[t-1,t]} = \Gamma_e^{[t-1,t]}\}$, $g \leftarrow \Gamma_1^{[t-1,t]}$
  if $|GP| \geq k$ **then**
    Mark all edges as anonymized, Update L and Γ
  end
  while $|GP| < k$ **do**
    Mark all Edges in GP as anonymized, Update L and Γ
    if $g[\gamma^{[t-1]}] == \Gamma_\gamma^{[t-1]}$ *and* $0 \leq \Gamma_\gamma^t < g[\gamma^t]$ **then**
      Cand_Edges=$\{x | x \in G^t and \Gamma^{[t-1,t]} = \Gamma_1^{[t-1,t]}\}$
      Find Cand_vertices for each edge $(u,v)\varepsilon$ Cand_Edges.
      Add Edges between (u,v) and $\ell \in CV$ to increase the nmf of (u,v).
      if *nmf(u,v)=g* **then**
        $GP = GP \cup \{(u,v)\}$, Update L and Γ .
      end
    end
    if $|CV| = \emptyset$ **then**
      Specialcase()
    end
    UpdateGS-Table()
  end
end
return $\bar{G}^t$, GS-Table

3.3. Incremental Updating of GS-Table.

When a new release of graph G^t comes, the GS-Table has to be updated and in every time period w, the GS-Table should be updated accordingly. So before attaching the edge information of G^t to the table, the edge information of $\bar{G}^{t-w}$ has to be removed and the edges will be re-sorted based on the number of mutual friend sequences $\Gamma^{w-1} = \gamma_e^{t-w+1}, \ldots, \gamma_e^{t-1}$ instead. After this removal, each edge will be in a k-anonymous sequence consistent group Θ_Γ^{w-1}. Therefore, there is no need to re-sort all edges and only the k-anonymous sequence consistent groups Θ_Γ^{w-1} need to be. This can reduce the sorting time. After that, the newly released graph $\bar{G}^t$ edge information will be appended to the GS-Table. The resorting of edges in each consistent group is also reduced. Hence, the appended GS-Table can be efficiently and incrementally updated. Fig 2 shows the pseudo code for incrementally updating of the GS-Table.

Example 5. *An illustration about update of the GS-Table follows. Considering Example 4, suppose a new edge i.e* $(5,6)$ *is added at time* t_3 *then the edges* $(3,5),(4,6),(2,6)$ *are changed to 4,2,2 respectively. Now, remove the information corresponding to time* t_1 *before attaching the yet to come information then re-sort the groups of edges, i.e.* $\{(1,3),(1,4),(1,5),(3,4),(3,5)\}$, $\{(1,2),(1,3)\}$, $\{(4,5),(2,5)\}$, $\{(1,7),(3,6)\}$, $\{(4,7),(4,6)\}$ *and* $\{(2,7),(2,6)\}$ *according to the number of mutual friend sequence* $\Gamma^{[2,2]}$. *Therefore, the order of* $\{(4,5),(2,5)\}$ *and* $\{(1,7),(3,6)\}$ *are changed. After that, the* t_3 *time edge information is added to* t_2 *information. Finally, the edges in the similar sequence consistent groups* $\Theta_\Gamma^{[2,2]}$ *are re-sorted according to the mutual friend sequence* $\Gamma^{[3,3]}$. *Table 1d shows the resulting table.*

Table 2: Anonymization Process

(a) t_1

(1,3),(3,8)	6
(5,8),(4,8),(3,5), (3,4),(1,8)	5
(3,7),(1,2),(4,5),(2,3), (6,8),(1,7),(1,4),(4,5),(7,8)	4
(5,6),(4,7),(2,8),(2,5), (3,10),(3,6),(1,10),(4,6),(7,10)	3
(8,9),(7,9),(2,10)	2
(6,9),(10,11),(10,12), (9,10),(11,12)	1

(b) t_2

(1,3),(3,8)	$6 \to 6$
(5,8)	$5 \to 6$
(4,8),(3,5),(3,4),(1,8)	$5 \to 5$
(3,7),(1,2),(4,5),(1,4), (2,3),(6,8),(1,7),(7,8)	$4 \to 4$
(5,6)	$3 \to 4$
(4,7),(2,8),(2,5),(3,10), (3,6),(1,10),(4,6),(7,10)	$3 \to 3$
(8,9)	$2 \to 3$
(7,9),(2,10)	$2 \to 2$
(6,9)	$1 \to 2$
(10,11),(10,12),(9,10),(11,12)	$1 \to 1$
(5,9)	$0 \to 2$

(c) t_2

(1,3),(3,8)	$6 \to 6$
(5,8),(4,8),(3,4)	$5 \to 6$
(3,5),(1,8)	$5 \to 5$
(2,3),(1,2),(4,5),(1,4),	$4 \to 5$
(3,7),(6,8),(1,7),(7,8)	$4 \to 4$
(5,6),(2,8),(2,5)	$3 \to 4$
(4,7),(3,10),(3,6),(1,10),(4,6),(7,10)	$3 \to 3$
(8,9)	$2 \to 3$
(7,9),(2,10)	$2 \to 2$
(6,9)	$1 \to 2$
(10,11),(10,12),(9,10),(11,12)	$1 \to 1$
(2,4)	$0 \to 4$
(5,9)	$0 \to 2$

(d) t_3

(1,3),(3,8)	$6 \to 6$
(5,8),(4,8),(3,4)	$5 \to 6$
(3,5),(1,8)	$5 \to 5$
(2,3),(1,2),(4,5),(1,4),	$4 \to 5$
(3,7),(6,8),(1,7),(7,8)	$4 \to 4$
(5,6),(2,8),(2,5)	$3 \to 4$
(4,7),(3,10),(3,6),(1,10),(4,6),(7,10)	$3 \to 3$
(8,9),(2,10)	$2 \to 3$
(7,9)	$2 \to 2$
(6,9),(10,12)	$1 \to 2$
(10,11),(9,10),(11,12)	$1 \to 1$
(2,4)	$0 \to 4$
(5,9)	$0 \to 2$
(2,12)	$0 \to 1$

(e) t_2

(1,3),(3,8)	$6 \to 6$
(5,8),(4,8),(3,4)	$5 \to 6$
(3,5),(1,8)	$5 \to 5$
(2,3),(1,2),(4,5),(1,4),	$4 \to 5$
(3,7),(6,8),(1,7),(7,8)	$4 \to 4$
(5,6),(2,8),(2,5)	$3 \to 4$
(4,7),(3,10),(3,6),(1,10),(4,6),(7,10)	$3 \to 3$
(8,9),(2,10),(7,9)	$2 \to 3$
(6,9),(10,12)	$1 \to 2$
(10,11),(9,10),(11,12)	$1 \to 1$
(2,4)	$0 \to 4$
(5,9)	$0 \to 2$
(2,12),(7,14),(9,14)	$0 \to 1$

(f) t_2

(1,3),(3,8)	$6 \to 6$
(5,8),(4,8),(3,4)	$5 \to 6$
(3,5),(1,8)	$5 \to 5$
(2,3),(1,2),(4,5),(1,4),	$4 \to 5$
(3,7),(6,8),(1,7),(7,8)	$4 \to 4$
(5,6),(2,8),(2,5)	$3 \to 4$
(4,7),(3,10),(3,6),(1,10),(4,6),(7,10)	$3 \to 3$
(8,9),(2,10),(7,9)	$2 \to 3$
(6,9),(10,12)	$1 \to 2$
(10,11),(9,10),(11,12)	$1 \to 1$
(2,4),(5,9)	$0 \to 4$
(9,14),(7,14)	$0 \to 2$
(2,12),(5,14),(7,13),(9,15),(5,15),(13,14)	$0 \to 1$

3.4. Anonymization Process

In this subsection, the current release of graph can be anonymized only by edge addition. The edges are added but not deleted because removing an edge severely destroys the structural information of a graph than adding edges. Two approaches are introduced for anonymizing each edge of a graph. In the first approach, let e belong to an existing k-anonymous sequence consistent group by adjusting (increasing nmf value of edge e). The second approach is creating a new k-anonymous sequence consistent group by increasing the NMF value of other $k-1$ edges such that these edges can form a k-anonymous consistent group with edge e. As specified in [15] anonymization of mutual friend attack is more challenging than k-degree anonymity model. In the k-degree anonymity model, an addition of an edge can just increment the degree of the nodes connecting this edge. In mutual friend attack anonymization model, adding an edge can increment the number of mutual friends of many edges. So when for anonymizing the dynamic graph, Anonymized Triangle Preservation Principle is used. When adding new edge it should not affect the number of mutual friends of already anonymized edges. Algorithm 3 describes proposed algorithm.

Example 6. *Consider the dynamic release network in Table 2 with both w and k set as 2. Table 2a is the privacy preserving release at* t_1. *Now consider anonymizing the sequential release at* t_2 *in Table 2b First, select the highest number of mutual friends edge from GS-Table i.e.* $(1,3)$, *the only way is, to create a new two-anonymous sequence consistent group* $\Theta_{(6,6)}^2$. *The edge* $(1,3)$ *is anonymized together with edge* $(3,8)$ *without additional edges as shown in Table*

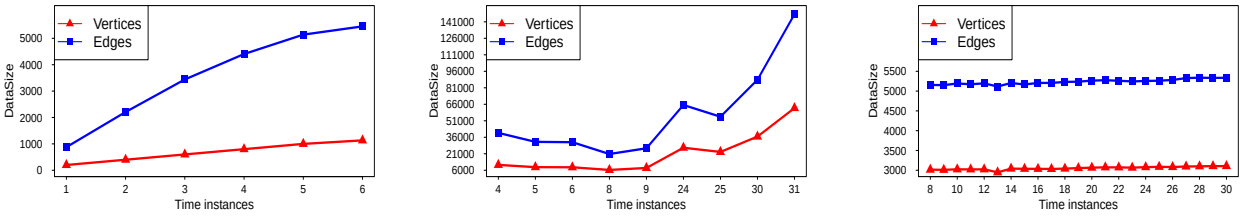


Figure 3: The no. of vertices and edges in each snapshot of (a) Email (b) Gnutella (c) AS-733 Datasets

2b. *Next for edge* $(5,8)$*, the only way to anonymize is to create a new* 2*-anonymous sequence consistent group* $\Theta^2_{(5,6)}$ *with candidate edge set Cand_Edges*=$\{(4,8),(3,5),(3,4),(1,8)\}$*, since no existing 2-anonymous sequence consistent groups* $\Theta^2_{(5,\alpha)}, \alpha \geq 6$*. Now we select the best candidate edge from the edge set, Cand_Edges to increase the number of mutual friend value which is equal to the edge* $(5,8)$*. By adding* $(2,4)$ *edge, the number of mutual friend values of* $(4,8),(3,4)$ *will be changed and the value is equal to the number of mutual friend value of the edge* $(5,8)$*. So the edge* $(5,8)$ *along with* $(4,8),(3,4)$ *will create a new* 2*-anonymous sequence consistent group* $\Theta^2_{(5,6)}$ *as shown in Fig 2c. After that edge groups* $\Theta^2_{(5,5)}, \Theta^2_{(4,5)}, \Theta^2_{(4,4)}, \Theta^2_{(3,4)}$ *and* $\Theta^2_{(3,3)}$ *are anonymized in the same way as edge group* $\Theta^2_{(6,6)}$*. For edge* $(8,9)$*, a new edge will be added i.e.* $(2,12)$ *to increase the number of mutual friend value of* $(2,10)$ *which is shown in Table 2d. The edge* $(7,9)$ *can be anonymized by merging already existing* 2*-anonymous sequence consistent group* $\Theta^2_{(2,3)}$ *since no existing* 2*-anonymous sequence consistent groups* $\Theta^2_{(2,\alpha)}, \alpha < 2$*. We then increase number of mutual friend value of* $(7,9)$ *by adding edges to fake vertex to form a new triangle as shown in Table 2e. Similarly the edge* $(2,4)$ *will be anonymized by adding fake vertices to increase the number of mutual friends of an edge* $(5,9)$*. Finally, all the edges are anonymized in the similar case of edge group* $\Theta^2_{(6,6)}$*. The anonymized release of* G^{t_2} *and the corresponding final resulting GS-Table is shown in Table 2f.*

4. Experimental Results

In this section, the performance of the proposed model is examined over the three real data sets [19] i.e. Email communication network from Enron, Gnutella peer-to-peer network and Autonomous system graphs (AS-733). The data sets are preprocessed into simple undirected graphs with out self loop and multiple edges.

Email communication network: This data set consists of email addresses posted to the web by the Federal Energy Regulatory Commission during its investigation. In this network, each node represents an email address and an edge represents a communication between nodes i.e. if at least one email was sent from one node to another node. In this 1,133 nodes and 5,452 edges are considered and the graph is divided into six partitions. The first partition is considered as the initial network and at every time stamp we add each partition to previous partition. So, there will be 6 sequential publications of this data set. The data size in terms of number of vertices and edges in each publication is shown in Fig. 3a.

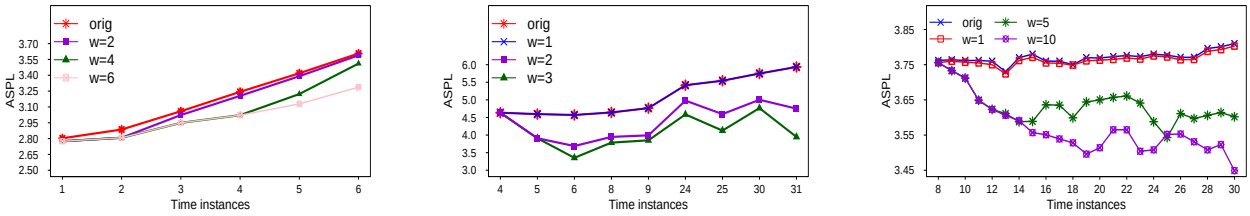
Gnutella: This data set is a peer-to-peer file sharing network which contains total of 9 snapshots collected in August 2002. Nodes represent hosts in the Gnutella network topology and edges represent connections between the Gnutella hosts. Fig. 3b shows the number of nodes and edges in each snapshot.

Autonomous systems AS-733: This is a communication network from the BGP (Border Gateway Protocol) logs. The data set contains 733 daily instances which span an interval of 785 days from November 8 1997 to January 2 2000. To conduct experiment we consider one day as time unit and have 23 daily instances from November 8 1997 to November 31 1997. Fig. 3c shows the detailed information in each snapshot. The AS data set exhibits both the addition and deletion of the nodes and edges over time.

The data sizes of email, Gnutella and Autonomous systems graphs are varied over time as shown in Figures (3a), (3b), (3c) respectively. The email network, number of vertices and edges are grow at each time instance but for Gnutella graphs the data sizes increase at some time instances and decreases in some other time instances. For autonomous systems, the increasing rate of vertices and edges slightly vary because this data exhibits both the addition and deletion of vertices and edges over time. Now we can observe the differences the way the graphs will evolve over a time will result in different efforts in anonymizing sequential networks for preserving the data utility. So the differences of data utility are plotted for anonymized graphs with respect to original graph. The performance results for email data set show a smooth curve while the curves of remaining two data sets will cross over each other at sometimes.

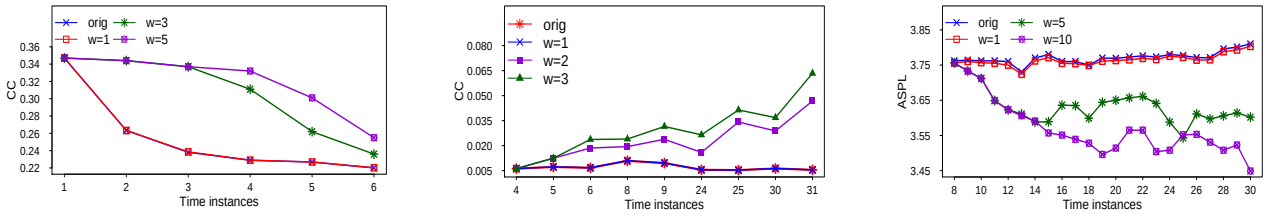
When the sequential released dynamic network is shared among third party sources it is difficult to monitor the release and who are accessing. The number of attackers and details of the attacker who accessed past releases is difficult. To predict the best value of w for compromising the level of the privacy and the data utility assumption is also impractical. We also need to consider whether the information disclosed in the past releases could be used to compromise the new releases or not in the privacy of dynamic sequential released networks. Therefore, the w value should depend on relevance of content over time. If the content relevance between the past and current releases are known and if it becomes small after $\bar{w}$ time units then the information disclosed in that very old version cannot be used to compromise the newly released version so the value of w could be set as $\bar{w} \leq w$ and $w \leq t$. To preserve the better data utility, the current snapshot is anonymized only with the content relevant network releases. But if the content relevance is high and rarely decays over time, the value of w will be increased and the anonymization of sequential releases will be the initial steps as specified in the proposed algorithm i.e., $1 \leq t \leq w$. This means the current snapshot is anonymized based on all previous releases. So there is a little chance that privacy leakage would happen if the value of w is larger but more the data utility in some cases which is shown in experimental results. Therefore, the data owner has to observe the content relevance over time and would take a decision on data utility and privacy guarantee.

The performance of proposed algorithm is evaluated in terms of the clustering coefficients (CC), average shortest path lengths (ASPL) and graph modifications under different protection mechanism



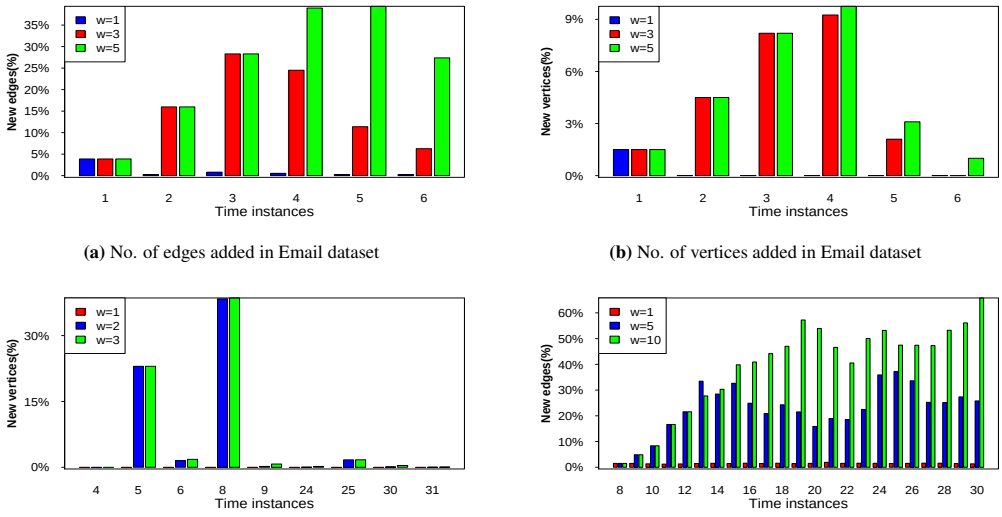
(a) **(b)** **(c)**

Figure 4: The effect of w on the average shortest path lengths (ASPLs) for a fixed k value: (a) Email (b) Gnutella (c) AS-733 Datasets



(a) **(b)** **(c)**

Figure 5: The effect of w on the clustering coefficients (CCs) for a fixed k value: (a) Email (b) Gnutella (c) AS-733 Datasets



(a) No. of edges added in Email dataset

(b) No. of vertices added in Email dataset

(c) No. of vertices added in Gnutella dataset

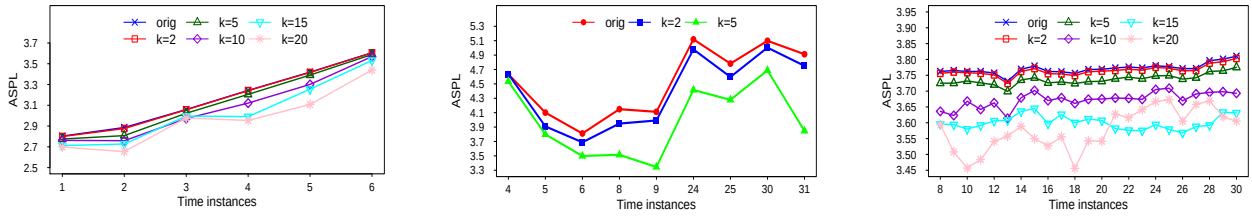
(d) No. of edges added in AS-733 dataset

Figure 6: The Graph Modifications with respect to different w settings

values i.e. w (different information hysteresis) and k (different protection level requirements). In all figures, the x-axis represents the time stamps of the new snapshot of the anonymized dynamic network with respect to the previous releases.

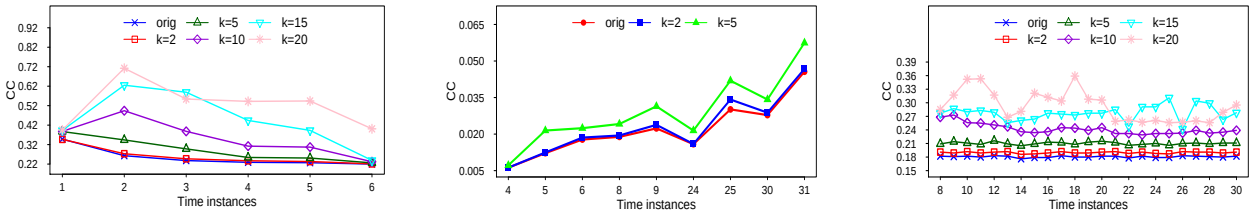
Protections against different information hysteresis (w): The performance results based on different w values under fixed k value are shown. For email data set, the k value is fixed as 5 and 2 for both Autonomous systems and Gnutella data sets. Figures 4a, 4b and 4c shows the average shortest path lengths (ASPL) for the anonymized and original graphs on three datasets. When $w = 1$, the ASPL values of anonymized graph are very close to original and changes in a similar trend to the original value. Because $w = 1$ is the base case to protect the privacy of the publication when it is static and the anonymization is based only on the current graph.When $w > 1$, the ASPL values of an anonymized graph biased some what more from the original graph values. Although, variations on ASPL values of the anonymized graphs are more stable compared to the base case of $w = 1$. Generally larger w leads to more information distortion. But, anonymization algorithm is based on not only the current graph, but also the $w - 1$ predecessors. So, larger w does not lead to the high distortion of information on ASPL values.

The performance result of clustering coefficients (CCs) of the anonymized graphs with the original graphs are shown in Figures 5a, 5b and 5c. We can see similar observations on CC values to the plots of ASPL. The CC values for anonymized graphs are close to the values of original graphs for the base case of $w = 1$. When $w > 1$, the anonymized graphs CC values deviates a bit more when compared with original graph CC values. Similar behavior is shown in both scale-free networks of anonymized graphs CC values. As the case of ASPL values, the information distortion on CC values need not be more when the w is larger. These results show that relation of content is also an impact factor for sequential networks so the setting of w has an impact, but does not govern the data utility preservation in anonymizations of sequential releases.



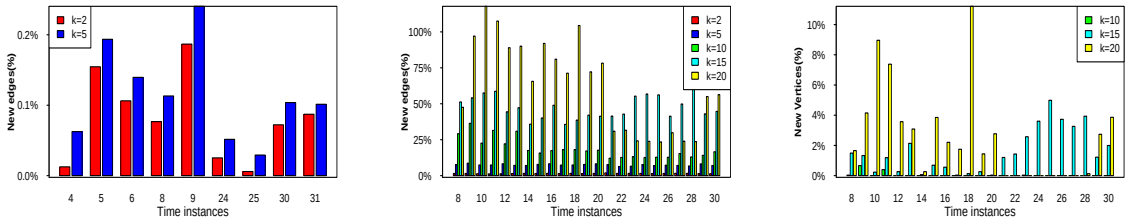
(a) **(b)** **(c)**

Figure 7: The effect of k on the average shortest path lengths (ASPLs) for a fixed w value : (a) Email (b) Gnutella (c) AS-733 datasets



(a) **(b)** **(c)**

Figure 8: The effect of k on the clustering coefficients (CCs) for a fixed w value : (a) Email (b) Gnutella (c) AS-733 datasets



(a) No. of Edges added in Gnutella dataset **(b)** No. of Edges added in AS-733 dataset **(c)** No. of vertices added in AS-733 dataset

Figure 9: The graph modifications with respect to different k settings for fixed w value

Figure 6 shows the how many graph modifications are involved. Dynamic k^w-NMF is an edge anonymization algorithm in which mostly we anonymize by adding edges between vertices not by deletion of edges and vertices. Figures 6a and 6d represents the number of new edges added for anonymizing email and Autonomous Systems data sets respectively. Figures 6b and 6c represent the number of new vertices added for anonymizing email and Autonomous Systems data sets respectively. The number of new vertices and edges added are almost zero for initial values of w because anonymization of sequential releases are based on previous releases. When w is varied from 1 to 3 and $w = 3$, for email data set and $w = 10$ when varied from 8 to 16 for Autonoumous systems the modifications involved keep increasing at the initial stage. After that, the graph modifications involved may increase or decrease as only the most recent $w - 1$ previous releases are considered. Thus, a larger w does not necessarily lead to more graph modifications.

Different protection levels requirement (k): The performance studies under different k values with fixed w value set as 2 for all the data sets are presented in this section. Figures 7a, 7b and 7c shows the ASPL values of the original and anonymized graphs. Generally a small k leads to less amount of information distortion. When the k value increases the information distortion also increases. But sometimes the ASPL values under large k value show less information distortion compared to smaller k values because the anonymization depends on previous releases not only the current graph. Figures 8a, 8b and 8c shows the CC values of the original and anonymized graphs. The CC values also show the similar observation i.e a larger k does not necessarily deviate more as of ASPL values, due to the previous sequential releases content dependency. Figures. 9a, 9b and 9c represent the graph modifications under different k values. The number of new vertices added are zero for all instances. As the k value increases, the number of graph modifications are needed. Generally, the above results show that the preservation of data utility is better when the value of k is small but the content dependence in sequential anonymized networks also has to be considered.

5. Conclusion

It has been shown that study of privacy preservation in dynamic networks should be focused. A privacy model, dynamic k^w-Number of Mutual Friend anonymity (k^w-NMF anonymity) algorithm for protecting edge identities sequentially released networks is proposed. To solve this problem, a heuristic technique is designed for anonymizing sequential released networks on three different data sets and developed a Group Sequence Table, the GS-Table, to summarize the edge information of sequential releases to improve the efficiency and the utility of the graph. The experimental results show that the proposed model can ensure the privacy of the dynamic network with very less distortion while preserving much of the characteristics of social networks.

References

[1] B. Zhou, J. Pei and W. Luk, "A brief survey on anonymization techniques for privacy preserving publishing of social network data", *ACM SIGKDD Explorations Newsletter*, Vol.10, No.2, (2008), pp.12-22.

[2] Wu, Xintao and Ying, Xiaowei and Liu, Kun and Chen, Lei, Managing and Mining Graph Data, *Springer US*,(2010).

[3] Cheng, James and Fu, Ada Wai-chee and Liu, Jia, "K-isomorphism: Privacy Preserving Network Publication Against Structural Attacks", *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, (2010), pp.459-470.

[4] Hay, Michael and Miklau, Gerome and Jensen, David and Towsley, Don and Weis, Philipp, "Resisting Structural Re-identification in Anonymized Social Networks", *VLDB Endowment.*, Vol.1, No.1, August (2008), pp.102-114.

[5] Liu, Kun and Terzi, Evimaria, "Towards Identity Anonymization on Graphs", *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, (2008), pp.93-106.

[6] B. Zhou and J. Pei, "Preserving Privacy in Social Networks Against Neighborhood Attacks", *2008 IEEE 24th International Conference on Data Engineering*, April (2008), pp.506-515.

[7] Zou, Lei and Chen, Lei and Özsu, M. Tamer, "K-automorphism: A General Framework for Privacy Preserving Network Publication", *VLDB Endowment*, Vol.2, No.1, August (2009), pp.946-957.

[8] Tai, Chih-Hua and Yu, Philip S. and Yang, De-Nian and Chen, Ming-Syan, "Privacy-preserving Social Network Publication Against Friendship Attacks", *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2011), pp.1262-1270.

[9] Bhagat, Smriti and Cormode, Graham and Krishnamurthy, Balachander and Srivastava, Divesh, "Prediction Promotes Privacy in Dynamic Social Networks", *Proceedings of the 3rd Wonference on Online Social Networks*, (2010), pp.6-6.

[10] N. Medforth and K. Wang, "Privacy Risk in Graph Stream Publishing for Social Network Data", *2011 IEEE 11th International Conference on Data Mining*, (2011), pp.437-446.

[11] Faloutsos, Michalis and Faloutsos, Petros and Faloutsos, Christos, "On Power-law Relationships of the Internet Topology", *ACM SIGCOMM Comput. Commun. Rev.*, Vol.29, No.4, August (1999), pp.251-262.

[12] C. H. Tai and P. S. Yu and D. N. Yang and M. S. Chen, "Structural Diversity for Resisting Community Identification in Published Social Networks", *IEEE Transactions on Knowledge and Data Engineering*, Vol.26, No.1, January (2014), pp.235-252.

[13] Zlatic, Vinko and Garlaschelli, Diego and Caldarelli, Guido, "Complex networks with arbitrary edge multiplicities", *Physics*, Vol.97, (2011), pp.8-11.

[14] C. H. Tai and P. J. Tseng and P. S. Yu and M. S. Chen, "Identity Protection in Sequential Releases of Dynamic Networks", *IEEE Transactions on Knowledge and Data Engineering*, Vol.26, No.3, March (2014), pp.635-651.

[15] Chong-Jing Sun and Philip S. Yu and Xiangnan Kong and Yan Fu, "Privacy Preserving Social Network Publication Against Mutual Friend Attacks", *Transaction Data Privacy*, Vol.7, No.2, (2014), pp.71-97.

[16] B. Zhou, J. Pei and W. Luk, "Anonymizing Sequential Releases", *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2006), pp.414-423.

[17] Xiao, Xiaokui and Tao, Yufei, "M-invariance: Towards Privacy Preserving Re-publication of Dynamic Datasets", *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, (2007), pp.689-700.

[18] Byun, Ji-Won and Sohn, Yonglak and Bertino, Elisa and Li, Ninghui", "Secure Anonymization for Incremental Datasets", *Springer Berlin Heidelberg*, (2006), pp.48-63.

[19] Jure Leskovec and Andrej Krevl, "SNAP Datasets: Stanford Large Network Dataset Collection", http://snap.stanford.edu/data, June, (2014).

[20] Vadisala Jyothi and V. Valli Kumari, "Privacy Preserving in Dynamic Social Networks", *Proceedings of the International Conference on Informatics and Analytics (ICIA-16)*, (2016), pp.79-86.