Securing Social Network Graph against Structural Attack based on Ant Colony Optimization

Munmun Bhattacharya
Department of Information Technology,
Jadavpur University, Kolkata, India

Bratati Hui
Department of Information Technology,
Jadavpur University, Kolkata, India

ABSTRACT
31% of global population use online social networks. These people can be from any class or may be very powerful, or may be world famous personnel, who are very likely to be stalked for inferring private information or people exploiting their privacy. For a better revenue, when any social networking sites sell these huge volume of data for either data mining or analysis to any third party organizations, the privacy of the users are put in danger. Naive anonymization is not effective enough to prevent most of the strategic attacks. Structural attack is kind of strategic attack. Here, using the structural knowledge available to the adversary, he can identify or infer some useful information about individual from a published anonymized social network graph. The available anonymization algorithms to prevent the structural attack are either insufficient or they degrades the data mining quality of the graph. So we consider preserving the data mining quality of the graph while eliminating most vulnerable edges under attack. We develop a privacy preserving strategy based on ant colony optimization which is a type of swam robot intelligence algorithm. We test this algorithm in some publicly available social graphs for checking the quality of the graph before and after anonymization using some known parameter of graphs.

General Terms
Online Social Network; Security; Internet of things.

Keywords
Social Network Graph;Perturb Graph;Anonymization; Structural Attack; Ant Colony Optimization.

1. INTRODUCTION
Behind every social networking site, there is a social network graph. A social network graph is a special graph, where nodes are the entities of the social network site and edges are the interconnections between those entities. These entities of the social network site can intern represent either individual user or organizations.

Over last few years, the number of social network users has grown exponentially. They upload a wild range of information, pictures, videos, comments, messages into social networking sites. This creates a huge data source. But, it is unfortunate that selling these data to the third party companies is the best way for the social networking sites service providers to earn revenue. In this way they make the data source visible to other peoples with whom the users do not want to share their information. These companies use those data for data mining or pattern analysis etc. But in wrong hand, that can be used to infer the information about users. The privacy of the users can be violated by disclosure of the identity of the individual user, sensitive relationships between two individual users and the data associated with each node inthe social network graph.

To preserve the privacy of the users, Naive anonymization is done on the graph. But most of the time, Naïve anonymization is not effective enough to prevent most of the attacks. Structural attack is a potential thread to the privacy of users of social network. Here the adversary tries to find out the identity of the target user from the published graph and based on the background information available to him. Random edge perturbation is not an enough option as it cannot prevent this attack when there is more than one edge with same edge weight and it cannotbe find out the most vulnerable edges in the graph.

In this paper, we have introduced a privacy preserving technique to prevent such kind of attack on social network graph. Our approach is based on the ant colony optimization. The anonymized graph maintains the performance of the original graph, which is more suitable for the companies and for the security of the user; it prevents re-identification based on the structural attack. Additionally, we demonstrate empirically that commonly studied structural properties of the graph, such as clustering coefficient, shortest path length, are barely changed by our anonymization procedure.

The remaining parts of this paper are arranged as follows: we will discuss about structural attack and some previous approaches to prevent this attack in Section II. Theoretical Problem is discussed in section III. Our algorithm and theoretical discussion are presented in Section IV. Experimental results are listed and discussed in Section V. Finally a brief conclusion is given in Section VI.

2. PREVIOUS WORK
At first, Backstrom et al. [2], explained how much vulnerable the privacy of the users of social network sites are under strategic attack. In their paper, they have discussed two types of attacks on social network graph: active attack, where the attacker attaches himself to the target node in the graph and in the published graph they finds this sub-graph and passive attack, where the adversary tries to find a known sub-graph connected to the target nodes. To construct the passive attack on the social network graph, the authors have discussed a type of structural attack. In this attack, the adversary colludes with some of his connected friends in the social network after the network is published and he builds a sub-graph, which can be identified in the published social network graph to compromise the privacy of their adjacent nodes.

Hay et al. [3] discussed a set of structural attacks. They discussed three types of structural attacks on the social network graphs: vertex degree based, structural signatures based and hub fingerprint based attack. In vertex degree based attack, they have shown how users can be identified from simple background knowledge like vertex degree and they

have given a formula of likelihood of an edge under a given knowledge query. For structural signatures based attack, they have shown a signature based checking whether there are existing sub-graph isomorphism, for measuring sub-graph knowledge. This iterative vertex refinement technique can work for almost all graphs. In addition, they have also described the hub fingerprint based attack on social network graph. Hubs are nodes with a high degree, high centrality and are highly visited. They can be easily identified from a published social network graph.

In a social network, hub fingerprint of a node is the list of hop count distances of the shortest paths from that vertex to hub vertices in that graph. To prevent such kinds of attacks, their approach was to divide the nodes of the whole graph into node partitions based on simulated annealing. After that, node partitions are replaced by a super-node, and edges between each pair of node partitions are replaced by a super-edge. Only the number of nodes in each partition is published along with the densities of the edges that are present within and across partitions. As the topological order of vertices is hidden from the adversaries, the vertex re-identification attack cannot violate the privacy of the users. But, their approach degraded the data mining quality of the graph. Experimental results show that our approach preserves the data mining quality better than other approach.

3. PROBLEM DESCRIPTION

A social graph consists a set of individuals, set of relationships between them and a huge volume of data share between them in the form of photo, video, and text etc. We calculate a new concept from the available information called degree of friendship in a social graph [1], which is basically measured according to the affinity of any two pair of the nodes. Any edge has got a particular weight depending on the affinity of two nodes towards each other. Now it will be represented mathematically.

Let G = (V, E, W) be a weighted undirected graph representing the original social network where V is the set of vertices, and each vertex or node represents an individual in the social network. E $\subseteq$VxV is the set of edges (relationships) between vertices, W stands for positive edge weights represented by variables x_1, x_2, ..., x_m (where each x_i corresponds to an edge i = (u, v) $\in$ E).

It is assume that the adversary knows a subgraph$G_s = (V_s, E_s)$ in the original graph G. Suppose the subgraphG_s contains N = $|V_s|$ no of nodes, with a list of vertex degree sequences known to the adversary; and the aim of the adversary is to identify G_s in G in the published data.

For anonymizing the original graph G, it is being tried to perturb the graph G which yields a perturbed graph *G'* = (V, *E', W'*) from the original graph by deleting some edges. At the time of removing edges from the original graph G, the less frequently used edges or the edges (u', v') where the weights (x_i) is minimum are selected. Both Dijkstra algorithm and ant colony optimization algorithm are used to find such edges with the highest possibility that the removing of those edges can secure the most vulnerable nodes under attack at the same time it will not degrade the data mining quality.

3.1 Dijkstra Algorithm

As Dijkstra's algorithm [7] is a well-known greedy algorithm for finding shortest paths for a single source to destination, this algorithm has been used to find all pair of shortest paths for the weighted social network graph. So that, during the use of Ant colony optimization for optimizing the results, it can be sure that the ants take shortest paths. This algorithm uses the natural behaviour of real ants. So the aim of this algorithm is to build a system that replicate the optimization technique of ants. This model is designed by replacing the real life parameters of a normal ant life with some useful technical parameters as inputs so that the resultant can be the optimized output. In this algorithm, natural habitat of ants is replaced by graph G, which we define earlier.

In Dijkstra's algorithm, there will be a start vertex v_0, at each iteration; the algorithm selects an edge e_i with the smallest known cost from v_0. If the edge e_i incident with vertex u, then the algorithm tries to find v, a neighbours of u with minimum cost by checking its cost, in the next step. In this way when the algorithm terminates, it finds shortest path (paths if exsists) P from one source v_0 to all other node in V which contains a subset of edges E in G.

Before placing the ants in the starting places, Dijkstra algorithm is used to find shortest path from the source vertex S to the destination vertex D. Next the ants are placed at the source vertex S, and the ants will then follow the path P which is found using Dijkstra algorithm. The algorithm then enters in its next phase Ant colony optimization.

3.2 Ant Colony Optimization

Ant Colony Optimization (ACO) is a way for designing metaheuristic algorithms [5] for combinatorial optimization problems. The Ant Colony Systems or the basic idea came from the real behaviour of the ants. The ants generally travel in a straight line to the food or nest. If an obstacle is placed between the nest and the food, to avoid the obstacle, initially each ant chooses to turn left or right at random. At the time of moving the ants will deposit pheromone in the trail uniformly. Now, if the ants that, by chance, choose to turn left will reach the food earlier, whereas the ants that go turning right around the obstacle and travel a longer path, to reach the food. Pheromone concentrates in the shorter path than the longer path around the obstacle due to the evaporation. Since ants have a preference to follow trails with more amounts of pheromone, all the ants gather at the shorter path around the obstacle.

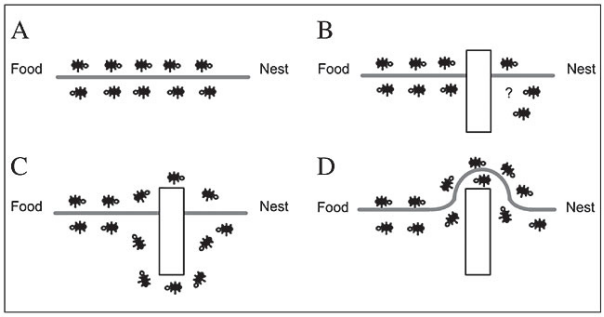

Fig 1: Real behaviour of ants

In this design, nest and foods are designed as the set of nodes in V in the graph G which are correspondingly the source node, S, and destination node, D. Ant agents are some artificial ants which try to find optimized path P from source to destination. The length from one node to another is called the distance, *Dis*. Dis_{ij} is the distance between nodes *i* to node *j*. The visibility is the reciprocal of distance, η. The visibility of the edge from node *i* to *j* is η_{ij}. Pheromone is replaced by artificial pheromone value, τ. Pheromone is like information written on every path, by an ant for the next ants to come. Paths with high pheromone value means an ant should choose

that path. Here the foraging behaviour is the random walk through graph (guided by pheromones). Initially, the amount of pheromone trail is equal along all the possible paths and is set to a very small value.

In ant colony optimization, updation of the pheromone value of an edge takes place when an ant visits that edge. If the said edge is chosen as the path of that ant, then the pheromone value of that edge increases and if the edge is visited but not chosen, then the pheromone value of that edge decreases.

To calculate the pheromone value, there are few parameters that need to be defined and set. In real world, each ant chooses next edge probabilistically according to the attractiveness and visibility. At first, the visibility (η) of an edge needs to be calculated. The visibility (η) is a heuristic function, which is chosen to be the inverse of the distance between two nodes.

$$\boldsymbol{\eta_{ij}} = \frac{\mathbf{1}}{\boldsymbol{Dis_{ij}}} \ldots (1)$$

The probability, p, is also called the evaporation rate, determines the probability of being chosen by an ant.

$$\boldsymbol{p} = \frac{[\boldsymbol{\tau_{ij}(e)}]^{\alpha} . [\boldsymbol{\eta_{ij}}]^{\beta}}{\sum [\boldsymbol{\tau_{ij}(e)}]^{\alpha} . [\boldsymbol{\eta_{ij}}]^{\beta}} \ldots (2)$$

The parameter α and β control the probability value. α is the history coefficient and it shows how important role the history plays in calculating the probability value. If α = 0, then it means the nearest nodes are much likely to be chosen. β is the heuristic coefficient and it shows how important role the pheromone trail plays in calculating the probability value. If β= 0, then it means only the pheromone amplification can modify the evaporation rate.

Finally, the pheromone is calculated using this following formula:-

$$\boldsymbol{\tau_{ij}(e)} = \begin{cases} (\mathbf{1}-\mathbf{p})\boldsymbol{\tau_{ij}(e)}, & \text{if the edge is not traversed} \\ (\mathbf{1}-\mathbf{p}).\boldsymbol{\tau_{ij}(e)} + \Delta\boldsymbol{\tau_{ij}^{k}}, & \text{if the edge is traversed} \end{cases} \ldots (3)$$

$$\Delta\boldsymbol{\tau_{ij}^{k}} = \frac{C}{L_k} \ldots (4)$$

where C is a constant and $L_k$is the length of tour of ant k.

4. OUR APPROACH

The social network graph that is published by online social network site owners is a representation of the online social network users and their interrelations present in the global stage in the form of a mapping that is widely called graph. This graph word is taken from graph theory, as the mapping of online social network reassembles a graph altogether with users as nodes and their interrelations as edges between those nodes. Before publishing the graph, at first naïve anonymization is done on the graph. In this process the names of the users are replaced by numbers or some generalized unique identifiers. But, unfortunately, naïve anonymization alone cannot ensure the complete security of the users. Further anonymization is needed to secure the data. We have used a combination of Dijkstra's Algorithm to find a tree with shortest path from all individual pair of nodes and Ant colony optimization for optimizing the resultant tree to find the edges which is used less frequently. Less frequently used edges are those edges where the affinity value [1] is very less. Theoretically as these links are contained less information, deletion of those edges from the graph will rarely effect.

Input :
"A social network Graph" file consists of a list of edges
K: percentage of edges you want to remove from the graph

Output:
A anonymized social network graph
Step by step actions of our algorithm
1. BEGIN 2. A social graph will be inputted to the system. From the inputted graph, all pair shortest paths will be found using Dijkstra algorithm. 3. A series of ants will traverse through every path found. 4. Pheromone is updated for every optional edge processed while finding the path but not traversed and for every traversed edge according to the equation 3, described in section 3. 5. When a colony of ants died, a new colony are unleashed, which learn from previous pheromone deposits of previous ant colonies. Then step 3 and 4 are iterated for n no. of times. 6. A subset of the edges having lowest pheromone value is chosen to be eliminated from the graph. 7. Depending on the value of K a set of edges,which is generated after step 6 will be deleted from the graph. 8. END

5. IMPLEMENTATION RESULTS AND ANALYSIS

For implementing this algorithm, a simulation model with the help of java and oracle as background database is created. A social graph G(V, E) has been taken as input. Randomly generated weights $W(x_0,x_1,\ldots,x_n)$ are added to all the undirected edges, where the values (Affinity values) are arbitrarily assigned in that graph. In this way a complete social graph G(V,E,W) has been created. Next Dijkstra algorithm is applied on the graph, which generates the set of all pair shortest paths, which are basically the paths from each node to every other node in the graph. Then ant colony optimization algorithm gets executed along all the paths, so that, edge deletion can be spread throughout the graph. From the resultant edge set of graph any subset can be chosen based on the criteria and delete those edges from the original graph.

For better experiment and to get original results, some publicly available social network graphs are used as dataset. The list of those datasets is given in Table 1.

Table 1. List of Real Graph datasets

Dataset	No of Nodes	No of Edges	Details
Dolphin Social Network	62	159	An undirected social network of frequent associations between 62 dolphins in a community living off Doubtful Sound, New Zealand, as compiled by Lusseau et al. (2003).
Zachary Karate Club	34	78	This is a classical social network dataset from the literature. The data was collected from the members of a university karate club by Wayne Zachary in 1977. Each node represents a member of the club, and each edge represents a tie between two members of the club.
Arenas Jazz Musicians	198	2742	This is a human social collaboration network between Jazz musicians, whose data was collected in 2003. Each node in the graph represents a Jazz musician and an edge denotes that two musicians have played together in a band.
American Football Network	115	613	This is a network of American football games between Division IA colleges during regular season Fall 2000, which is compiled by M. Girvan and M. Newman.

For measuring the effects of anonymization using this algorithm, two simple parameters Average Path length and clustering coefficient are used to describe the quality of the anonymization and anonimized graph. For measuring that, first both the parameters the Average Path length and clustering coefficient of the original graph are calculated and stored. Next, the above said algorithm is applied to generate the anonymizedgraph by removing edges. Theoretically, instead of random deletion of edges if a metaheuristic algorithm is used to remove the edges which takes the edge weight values into consideration will give the optimized result.

To calculate the change in Average path length and clustering coefficient, all paired shortest path of the original graph need to be saved into the database. Database tables contain the source, destination and all intermediary nodes of the path along with the length of the path. So that after anonymizationof the graph, changes can be calculated. For perturbation of the graph, a variable K (= {1, 2, 3, 4, 5}) which specifies the percentage of edges will be removed from the graph is used for anonymization. For better analysis, first 1% of the total edges, E are removed from the original graph, then 2%, 3% and so on and the effects of these changes on to the graph properties are recorded.

To analysis this algorithm, it has been applied on multiple datasets given in table 1.The results are in the following figures, where the effect of changes on the clustering coefficient and average path lengths of the graph against removing of percentage of edges from total number of edges has been shown. In the following figures blue line shows the value in original graph and red line describes the same parameter values after anonymization. Almost in all the cases, the changes between original graph and perturb graph are very less. Only in Fig 3, the difference of avarage path length between orginal graph and perturb graph is slight noticeble. The probable reason for it can be, if the edge weight are same for more than one edge then any one of these will be selected for deletion by this algorithm and the selected edge may be an impotant one for that shortest path. Other than this, all the results are fulfilling the expectation of this algorithm.

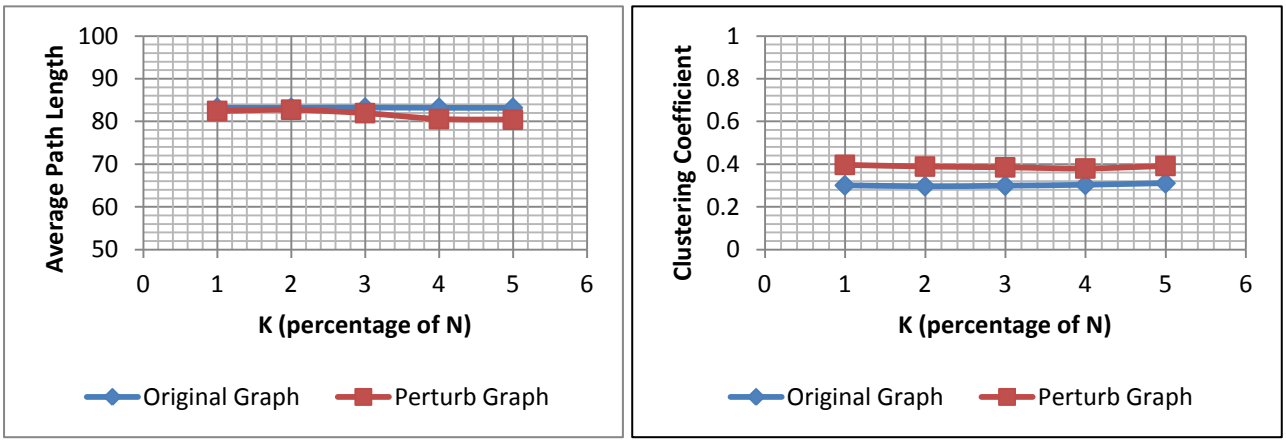

Fig 2: Effect on clustering coefficient and Average Path Length with changing K (Percentage of edges removed from the graph), in dolphin social network.

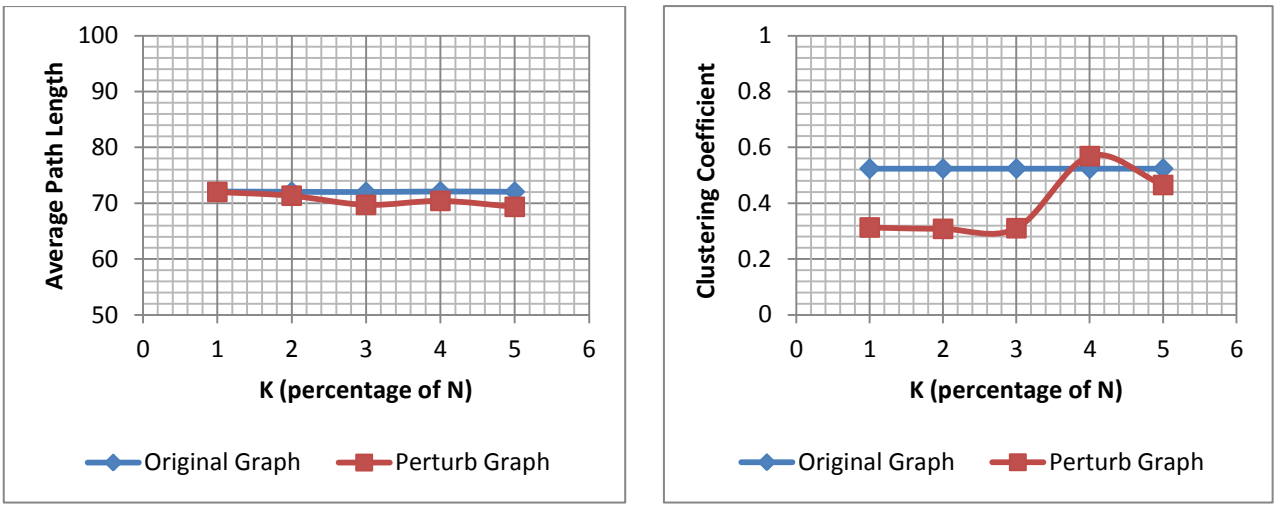

Fig1: Effect on clustering coefficient and Average Path Length with changing K (Percentage of edges removed from the graph), in Zachary Karate Club.

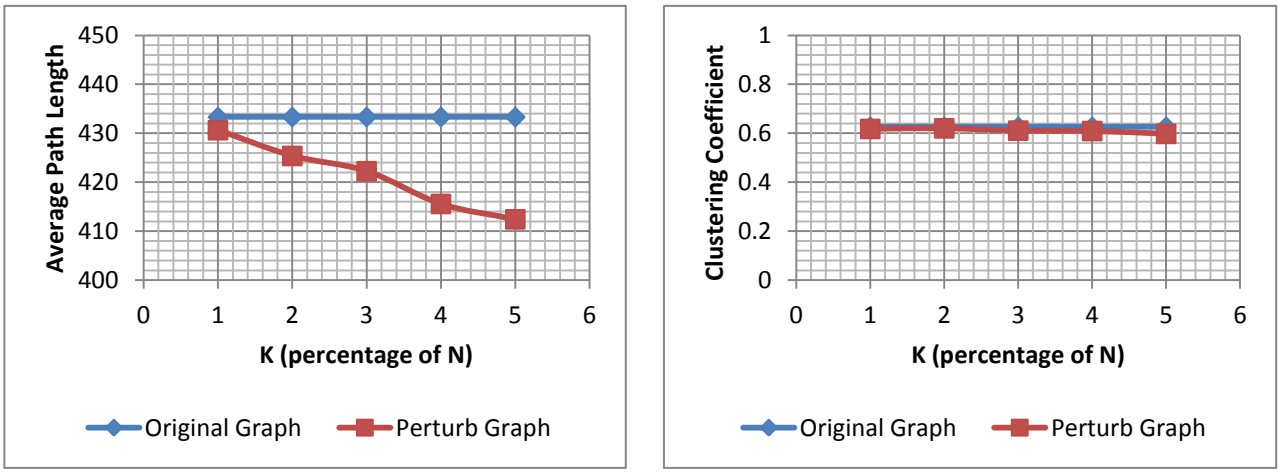

Fig 2: Effect on clustering coefficient and Average Path Length with changing K (Percentage of edges removed from the graph),in Arenas Jazz Musicians.

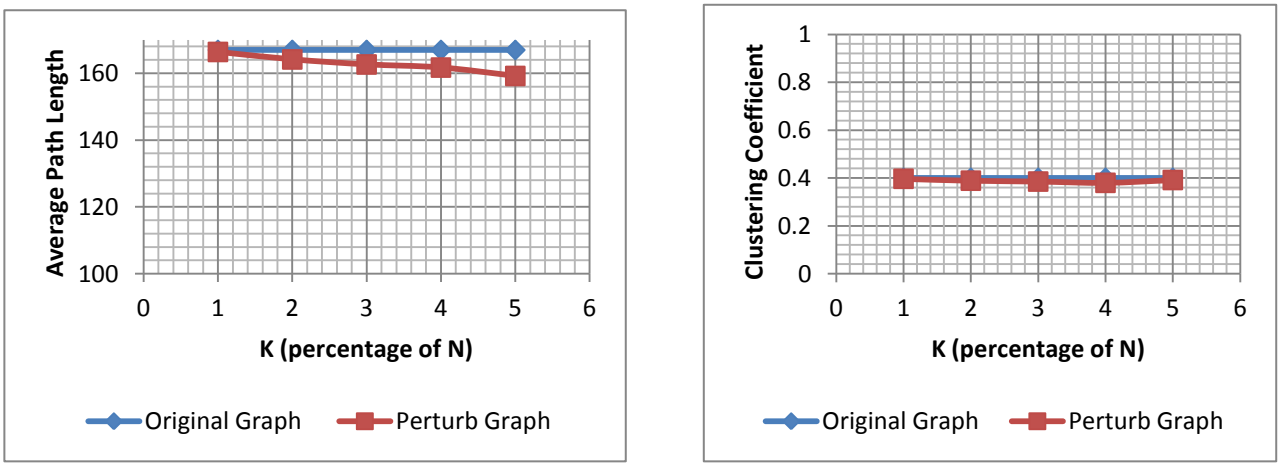

Fig3: Effect on clustering coefficient and Average Path Length with changing K (Percentage of edges removed from the graph), in American Football Network.

6. CONCLUSION AND DISCUSSION

There is a high privacy risk in published social networks data in terms of attack, identity also can be revealed. This paper has tried to identify some of them. A special type of passive attack, called the structural attack, is the main consideration of this paper and it has proposed a new concept of graph anonymization technique with the help of Dijkstra Shortest Path Algorithm and Ant Colony Optimization to protect the graph against such attacks. For finding optimal solution using ant colony optimization, at first Dijkstra algorithm is used to find a spanning tree with all vertex, so that ant colony optimization theory could be apply on this tree. This algorithm is design to use on a static graph but as it maintain a complete graph which will give the scope to use this algorithm even if new nodes are getting connected (by friendship) in the network. It is seen that this algorithm perturbs the graph, efficiently tries to prevent structural passive attacks, with maintaining the data mining quality of the social graph. The resultant graph generated by this algorithm maintains the parameter values of the original graph, which is more suitable for the companies, the security of the users; it also prevents the structural attack from occurring. The experimental results also demonstrate the facts that this approach can preserve the data mining quality of the original graph as same as possible and try to reduce the scope of attacks.

This algorithm is designed for the small static graph, where no. of vertex, edges and edge weights are fixed throughout the execution. But in real world social networks are dynamic. People are joining or escaping from the network changing the structure of the network. Even with every like, post etc. the edge weight will change. This algorithm can be modified in future for dynamic network. Only some small real graphs are used to test the algorithm, but it only claim the credits if it will work for large real life graphs. Only average path length and clustering coefficient are used as graph quality measuring parameter but there are so many left to check which can be tried in near future. This algorithm changes the graph characteristics a little. So minimize the effect of this change, instead of deletion of edges, selected edges can be shift from one path to another path. At the time of shifting edges, vertex degree should be taken in consideration, so that it cannot be detected.

There are different types of passive attacks are available; this work can also be amalgamated with the relational anonymization. So that it can nullify many attacks at a single go. Any other heuristic optimization algorithm also can be used to solve this problem.

7. REFERENCES

[1] Munmun Bhattacharya, Sandipan Roy "Prevention of Walk Based Attack on Social Network Graphs Using Ant Colony Optimization" IEMCON 2015, Vancouver, Canada, October 15th -17th, 2015.

[2] Lars Backstorm, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography. In International World Wide Web Conference, 2007.

[3] Michael Hay, Gerome Miklau, David Jensen, Don Towsley, and Philipp Weis. Resisting structural re-identification in anonymized social networks. In International Conference on Very Large Data Bases, 2008.

[4] Zhou, Bin, and Jian Pei. "Preserving privacy in social networks against neighborhood attacks." 2008 IEEE 24th International Conference on Data Engineering. IEEE, 2008.

[5] Dorigo M. and G. Di Caro (1999). "The Ant Colony Optimization Meta-Heuristic", New Ideas in Optimization, McGraw-Hill, 11-32, ISBN:0-07-709506-5.

[6] L. M. Gambardella, ` E. D. Taillard, and M. Dorigo. "Ant colonies for the quadratic assignment problem". Journal of the Operational Research Society, 50(2):167–176, 1999.

[7] E. W. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik, vol. 1, pp. 269–271, 1959.

[8] Das, Sudipto, E. Ömer, and Amr El Abbadi. "Anonymizing weighted social network graphs." 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010). IEEE, 2010.

[9] Mislove, Alan, et al. "Measurement and analysis of online social networks." Proceedings of the 7th ACM SIGCOMM conference on Internet measurement. ACM, 2007.

[10] Lusseau, David. "Evidence for social role in a dolphin social network." Evolutionary ecology 21.3 (2007): 357-366.

[11] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt and A. Arenas, Physical Review E , vol. 68, 065103(R), (2003).

[12] M. Girvan and M. E. J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99, 7821-7826 (2002).