

Community privacy preservation

Prof. Qiang Qu
Innopolis University
Kazan, Russia
Email: q.qu@innopolis.ru

Dr. Sadegh Nobari
Innopolis University
Kazan, Russia
Email: nobari@innopolis.ru

Usvyatsov Mikhail
Innopolis University
Kazan, Russia
Email: m.usvyatsov@gmail.com

Abstract—The abstract goes here.

I. INTRODUCTION

Nowadays there is a lot of information that cannot be distributed as is. Increasing popularity of social networks, p2p networks and other communication networks motivates research on the restrictions that should be applied to the data that is available.

The most common representation of a network is graph. Graph represents the structure and relationships between objects and thus, it should be preprocessed before it can be published due to privacy restrictions that do not allow publish such information as is. There are a lot of works that study graph anonymization problem. This method proposes the property that no one node in a graph can be distinguished with the probability 1. Although it prevents the information about users interconnections to be distributed, knowing the community structure of the network it is possible to distinguish one community from another. It means that the person itself cannot be compromised but all the group of people can be.

The goal of this work is to propose the method to maintain anonymization property of graph itself and moreover to keep the community structure to be anonymized.

A. Preliminaries

We define network as $\langle V, E \rangle$, where V is a set of graph vertices and E is a set of edges between these vertices. According to [1], k -degree anonymity is the property that each vertex in a graph has at least $k - 1$ other vertices with the same degree.

The problem of graph anonymization is solved and one of the solution is FGD algorithm defined in [2].

Our goal is given the graph G transform it to G' to make it not only k -anonymous but also its community graph k_1 -anonymous.

Division of graph into communities is a very big problem. First of all the definition of community is very inaccurate. For example, every clique is obviously a community, but not every community is a clique. Different algorithms of communities detection use different interpretation of a word "community". Newman introduced the idea to use modularity as an objective function in [3]. And in [4] COMBO algorithm was introduced. It optimizes the function of graph modularity and tries to perform three possible operation: merge two communities into one, divide the community into two and to move nodes from one community to another until the modularity gain grater a very small limit. In another words, while algorithm converges the modularity change is big from iteration to iteration.

After the division initial graph into communities there are several ways to build community graph. Each of them incorporates different opposition attack model. The first community graph construction can be done via algorithm 1.

Data: communities - list of communities sorted in order of n vertices definition

Result: G - community graph

G = new empty graph

D = empty dictionary

for $i = 1$ to n **do**

 | add $vertex_i$ to $D[communities_i]$

end

for $community_i, community_j$ in each of

$C^2_{communities}$ **do**

 | **for** $edge$ in $dot_product(D[community_i],$

$D[community_j])$ **do**

 | $G.add_edge(community_i, community_j)$

end

end

return G

Algorithm 1: Build communities multi-graph

Such graph will have a lot of edges between several communities because for every communities interconnections it will create a new edge between this communities. On the other hand algorithm 2 builds another form of community graph that makes an edge only once.

Thus single-graph can be defined as a graph where two vertices have at most one edge between them and multi-graph is not limited by the number of edges between each two nodes.

II. CONCLUSION

The conclusion goes here.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

- [1] James Cheng, Ada Wai-chee Fu, and Jia Liu. K-isomorphism: privacy preserving network publication against structural attacks. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 459–470. ACM, 2010.
- [2] Xuesong Lu, Yi Song, and Stéphane Bressan. Fast identity anonymization on graphs. In *Database and Expert Systems Applications*, pages 281–295. Springer, 2012.

Data: communities - list of communities sorted in order of n vertices definition

Result: G - community graph

G = new empty graph

D = empty dictionary

edges = empty dictionary

for $i = 1$ to n **do**

 | add $vertex_i$ to $D[communities_i]$

end

for $community_i, community_j$ in each of

$C^2_{communities}$ **do**

for $edge$ in $dot_product(D[community_i],$

$D[community_j])$ **do**

if $(community_i, community_j)$ not in edges

then

 | edges[$(community_i, community_j)$] = 0

end

 edges[$(community_i, community_j)$] += 1

end

end

for $i = 1$ to $edges.length$ **do**

 | G.add_edge($edge_i$)

end

return G

Algorithm 2: Build communities single-graph

- [3] Mark EJ Newman. Communities, modules and large-scale structure in networks. *Nature Physics*, 8(1):25–31, 2012.
- [4] Stanislav Sobolevsky, Riccardo Campari, Alexander Belyi, and Carlo Ratti. General optimization technique for high-quality community detection in complex networks. *Physical Review E*, 90(1):012811, 2014.