Identity disclosure of Anonymized Social Actors in the Network using Seed then Grow model.

Chithra Apoorva D A[1], Nandini S[2],Mala B M[3],Dr.Brahmananda S H[4]
[1] Assistant Professor, Dept of CS & E, GITAM School of Technology, Bangalore.
[2] Assistant Professor, Dept of CS & E, GITAM School of Technology, Bangalore.
[3] Assistant Professor, Dept of CS & E, GITAM School of Technology, Bangalore.
[4]Professor, Dept of CS & E, GITAM School of Technology, Bangalore.

Abstract: Social network is a set of nodes and edges interconnected to form a structural graph, here the node represents the social actor or the user who may be individual or any organisation and the edges represents the relationship between the social actors. The footprints left behind by the social actors on the anonymization social graph leads to data hack. To notify the social actors whose data has been hacked, we use an algorithm Seed then grow. This algorithm helps in the identification of social actors from the anonymised graph. Identification is done by the structural similarities in the social graph. Sub graph is known as the seed. Initially the seed is implantation by attacker on the social structural graph. Then the seed is made to grow based on the similarities found in the network. The seed then grows to identify the social actors and notifies regarding the hacker and by removing the arbitrary parameters of the previous work. This algorithm has very little adverse information and is efficient, also accurate.

Key terms: Social Actors De-Anonymization, Privacy**,** Hack.

1 INTRODUCTION

In the current world the social media is the fastest media than the television, newspapers. Social media are the platform which allows actors of social network to share images, videos, and new ideas. As eBizMBA site says the top three social networking sites Facebook, Twitter, LinkedIn. Approximately there are 90 crore of monthly visitors to Facebook, 31 crore for Twitter and 25 crore for LinkedIn. Alexa Ranks 2nd, 8th and 9th position respectively in March 2015. [A system which is a measure for Ranking is Alexa. Ranking is based on the frequency of the views on websites. The data traffic rate recorded by the alexa.com over a three months of duration is used to calculate the rank.]Gaming applications, photos, videos shared on the social media drags the interest of the social actors. The digital footprints of the social actors over a social media website allows the web tracker like trackur, google analytics etc., to gather the personal information.

Third party actors may be like gaming applications or websites gathers personal data of the social actors. Based on the data what the third party actors collected, they post advertisements and post value added application services for the intendant social actor. One of the top social networking Facebook's privacy policy tells- when a social actor logins to Facebook from any devices like computer, mobile. It collects cookies, personal information of the user which are decided to not to disclose. Also website tracker gathers the data like IP address, Browser and wed pages to which user view. To make better advertisement on Facebook these information is very much useful. The personal data are collected by the website tracking tools like trackur, crazyegg, google annalistic etc. Because of all the above constraints and also because of privacy breaches, it's a major dealing of data storage, data processing and data publishing over the social network. A very simple and basic idea to preserve privacy is to remove the label of the nodes on the network. Labels like name, identity, email and other identities.

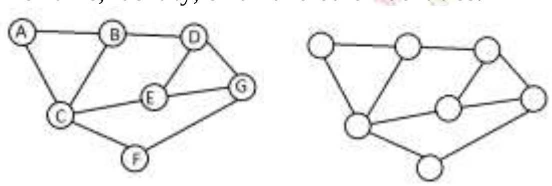

Fig.1, tells anonymization

Here comes the question that after the anonymization of social graph, the privacy is really maintained? L.backstrom [3], in his work, states some types of attacks against compromising social actors privacy And A.Narayanan's [2] work tells that only anonymization is not enough for maintaining privacy. The research made till now are still infancy and need to be more up to the mark. In this paper, we propose a two stages of identification that is Seed then Grow. The attacker or the hacker initially implants a seed into the targeted network. After the anonymization of the network, the seed grows and hence reaches target social actor and retrieves data from it. This is how the privacy is compromised.

The proposed algorithm mainly concentrates on the following.

Effective seed construction: The Attacker does not have the complete information of the graph except the seed. The seed is a star graph. Except the star nodes, the attacker has no information about the nodes in the graph. The seed and its information is only available to the attacker.

Seed recovery: The seed recovery algorithm has the information of two – hop neighbour nodes in the social graph and that's how it is efficient.
Seed grow: In the proposed algorithm, seed is made to grow, which means the target users are identified and that's how the privacy is lost. Previous works used arbitrary parameters and this algorithm has maintained a fine balance between accuracy and efficiency.

2 RELATED WORK

Social network can be mathematically represented in the form of graph. A social network graph known as G has its vertices V which represents social actors and edges which connects the vertices, mathematically can be written as E C V×V. For the social graph, the vertices and edges are labelled as social actors' relationship respectively.

In the previous work [2,3] privacy was maintained by removing the label of edges and vertices, the process of removing labels of a social networking graph is known as anonymization. The extended work for privacy preserving is modelled in terms of centrality [Centrality which means identification of the vertex which is more important in the network.]. This idea is taken by the 'Social Network Analysis' [9].To maintain privacy only by removing names and social security number are not sufficient. Databases can have the similar attributes which may help in the re-identification of the social actors. Similar attributes may be Date of birth, gender, zip code [10].L.backstrom, C.Dwork et al [3] explored number of attacking models and proved compromising privacy and also introduced a term 'structural steganography'. The other previous works [5, 6, 8, and 11] are showing how privacy is being compromised by the data's *utility* and the *background knowledge* of the adversary.

Utility of the social graph's data is removing randomly the edges or adding edges randomly by keeping the nodes as it is. How much the published graph is distorted, that much less useful it is [4]. The other previous work [5, 4, 6, 8 and 11] are all using an ideology that by varying the utility of the published graph, the re-identification of the social actors may be reduced. Besides, changing the usefulness of the network, it's hard to prevent the data attack. In current days, the online social networking sites provide APIs to facilitate development of third party application. These application programming interfaces can be subjected by the malicious hacker to gather the data in the social network.

Background knowledge can be said as the information about the target node which leads to the privacy compromise [12]. Gathering the background knowledge by the adversary is not only restricted to the target's neighbour node. The adversaries' knowledge may be modelled in identifying attributes of vertices, cost of the link, and labels of the edges, graph matrices , degree of vertex and relationship of links, neighbourhoods, embedded sub graph and also the adversaries knowledge may span many other networks, including target alter network[2].
This is the real assumption that the user uses more than one network services. For example Bob Marley uses Facebook and also other complimentary services like flicker. It is very common that the user of one service would use another service at the same time. As the user registers to another social networking service, his relationship in those network might be same in the first social network, which may leads to leak of valuable information for the attacker and this similarities of the relations in two different social network services provider like Facebook and flicker are the threat for privacy. The above observation inspires seed and grow algorithm.

[Motivation scenario] Consider Marley is an employee of f-network and he maintains the database of f-net. Marley becomes eager to know who the actors in the f-net are. He will check out the other network service providers like g-net and somehow he will identify four actors of the g-net. By the structural similarities of g-net, he will be successful to identify the four actors in the f-net also. And also he will be successful to identify 100 more actors from the anonymised graph.

We conclude with a comment on our model. The de-anonymised attack on the target social actors uses undirected graph. The idea of undirected graph is arise naturally by the scenario where the social relationship of the actors is mutual, which means, a friend request sent must be accepted to make an edge as undirected. Here comes another scenario where a fan follows his favourite celebrity on twitter. In this situation the relationship is not mutual. The undirected graph is the special case of directed graph. The proposed algorithm works in the same way for both directed and undirected graph. For the ease of use undirected graph is considered.

3 HACK BY USING SEED THEN GROW ALGORITHM.

Here is how the users in the social graph are identified. Let us consider a graph $G_T = \{V_T, E_T\}$ which represents the target social network after removing the ID's of the user ,that is after anonymization and also, Let us consider a graph $G_B=\{V_B,E_B\}$, which represents the background network, which has been constructed by the attacker using the background knowledge which he has. The motivation scenario tells about how this G_B graph is constructed. The so called hacker's goal is to identify the vertices V_T in the target graph by considering the background graph and the structural similarities between G_B and G_T. Assuming that, user's profile which belongs to same user has the same relationships in both G_B and G_T. The structural similarities are affected by the sporadic connections which are made from stranger on either of the G_B or G_T [13]. Such sporadic relationship can be removed [13] by quantifying the connection's strength. The remaining network has the strong relationship which shows the user's real-world social relationships, which gives birth to the identifying the structural similarities of G_B and G_T and hence the graph G_T and G_B are syntactically same but semantically different, which means the graph connections looks same but the meaning associated with those are different. The vertices in the target graph G_T are re-identified with the help of background graph G_B. And hence the privacy is compromised.
We assure that the attacker does not has the complete control the over the target graph But somehow by the theft of the user profile and once the user profile is attacked it is said as the initial seed . the efficiency of the seed implantation is based on the Sybil detection

of forgery attack [14,15,16,17,18,19,20,21,22]. In our algorithm as the number of initial seed are increased the capability of identifying or deanonymization of the nodes are done quickly. The proposed algorithm is of two stages-Seed then grow is mainly using the structured based vertex matching.

Seeding and Recovering: An imitation graph to target graph is made which can be called as G_F, finger print graph, where G_F belongs to G_T. G_F is the sub graph of G_T. After the anonymization of G_T is done and published, the G_F is recognised in the G_T. The vertices V_S belongs to G_F in G_T are known as the initial seed and the seed is made to grow.

Growing: Once the initial seed is planted over the G_T with the help of G_F, the seed is made to grow, which means, the one hop neighbours nodes to the G_F in G_T is identified and This identification is looped until all the user in G_T are identified

3.1 Seed

3.1.1 Efficient seeding of G_F on G_T requires the following two graph structural properties.

- Only one G_F graph should be identified on G_T for example,

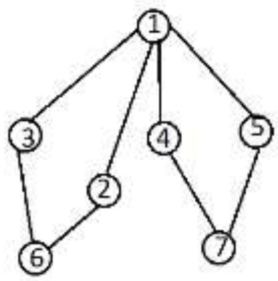

Fig 2, Randomly generated graph.

Consider the above Figure 2 graph, Once the labels are removed the subgraph of vertex set {1, 2, 3, 6} and {1, 4, 5, 7} are identical. Identifying these kind of sub graphs G_F a G_T may lead to ambiguity and hence G_F should be uniquely identifiable on G_T.

- Asymmetric subgraph of G_F should be identified on G_T for example, in the Figure V_1, V2, V3, V6 are symmetric to V1, V4, V5, V7. Therefore identification of G_F should be not having automorphism.

The randomly generated finger print graph is supposed to be uniquely identified on the G_T, and it may not satisfy the asymmetric Property. Since the main aim of seed is to identify the user rather than identifying finger print graph [finger print graph which satisfies the non-isomorphic and asymmetric nature of graph] therefore requirement of asymmetric graph of G_H can be flexible. For the pair of vertices u belongs to V_S, Let us consider $V_F(u)$ be the vertices in the finger print graph which Connects to u. For all pair of vertices, like u and v in the V_S, where V_F (u) and $V_F(v)$ are always distinguishable in Finger print graph G_F ,which means , the sequence degree of the $V_F(u)$ and $V_F(v)$ should be different. More clearly, the property – automorphism should not be there. In the Figure 2, $V_F(6)$={V3,V2} VF={V4,V5} are not uniquely identified on G_F. By all these observation we propose an algorithm for constructing and recover of finger print graph G_F.

3.1.2 Construction of seed

Initially the G_F is identified on G_T with a star structure. The centre node of the star structure is known as the v_h vertex head [also can be called as head vertex] of GF. V_h only connects to all the nodes which are very next to the head node in G_F.

All the vertices expect the head vertex are connected with one or the other vertices of the initial seed V_S in the target graph. To confirm the seeds u and v are not unique, the attackers can deny the connection requests in the target graph, which are $V_F(u)=V_F(v)$. Notice that the Attackers has won't be having all the control over the social network.

After the identification of the star graph on the target graph, the attacker constructs other connection with the G_F. Two properties to do this are

1. When u and v are the two initial seed, G_F should not map $V_F(u)$ to $V_F(v)$ which means no automorphism.
2. The established G_F should not have any unique structural patterns for anyone except the attacker.

The first principle follows section 3.1.1 that is unambiguous pair of initial seeds u and v are to be identified only if no automorphic $V_F(u)$ and $V_F(v)$ maps. The second principle has a dilemma G_F should jumble with the rest target graph, yet to be distinctive. In this discussion first we justify the first principle and the will resolve the dilemma. The idea behind jumbling the rest of the nodes on G_F is to avoid the distinct structural patterns which are help full for the anti-attackers. If the finger print graph is not jumbled, may be the defence may for the attack would be done over the GF by the pattern matching mechanism. An implication is that the construction of G_F should be stochastic rather than deterministic. Yet, stochastic construction alone is not enough for G_F to blend into G_T . Numerous studies [25, 26, 27, 28, 29, 30, 31] indicate the existence of distinctive structural properties of online social networks as opposed to arbitrary random graphs. In particular, online social graphs consist of a well-connected backbone linking numerous small communities [25].Within each community, vertices show a local, transitive, triangle-closing connection pattern [29]. The construction of G_F should reflect these properties to blend into G_T . The cost for the attacker to establish the finger print graph is more, because of the number and the various connection patterns in between the V_F and V_S initial seed. To minimise cost for establishing G_F should mimic a local community network G_T [25].After constructing the star structure with the head vertex vh at the centre, all the pair of vertices in V_F-{v_h} Are connected with the probability t, where the probability t is the transitivity of the community network in G_T, likely to say, the two vertices having a same neighbour (v_h in G_F) will be having a connection to each other. Practically, always the attacker will be knowing auxiliary information about the target graph G_F, Also we can tell he will be having information about the community transitivity and community size. The establishment of G_F should be adjusted to such auxiliary information for G_F to fit it on G_T, After the rest of the vertices in V_F i.e V_F-{v_h} is connected with a probability of t, the attacker find the internal degree $D_F(v)$,which means the node of vertices which are connected to v in V_F and is ordered in a increasing sequence S_D. For every v belongs to V_S, v has corresponding subsequence $S_D(v)$ of S_D. For example, V_6 has a connection to v_2 and V_3 from G_F since degree of V_2 and d of V_3 is 1, sequence degree is <1,1>. If $S_D(u)$ not equal $S_D(v)$ for u and v in V_S. there will be no automorphism which will map to $V_F(u)$ to $V_F(v)$. this is how the unambiguous of connection is overcome. If the property of ambiguous is not satisfied, the attacker repeats the random connection in the G_F(except the head vertex) until the unambiguous graph connectively is obtained. The v_h, S_D and V_S are the secrets gathered by the attacker. All these combing helps to recover G_F from G_T. The combination of secrets give the high probability to recover the G_F unambiguously from the anonymised G_T.

TABLE 3.1: Algorithm of seed construction.

Create $V_F = \{v_h, v_1, v_2, . . .\}$.
Given connectivity between V_F and V_S.
Connect v_h with v for all $v \in V_F - \{v_h\}$.
loop
for all pairs $v_a \neq v_b$ in $V_F - \{v_h\}$ do
Connect v_a and v_b with a probability of the community transitivity t.
end for
for all $u \in V_S$ do
Find $S_D(u)$.
end for
if $S_D(u)$ are mutually distinct for all $u \in V_S$ then
return
end if
end loop

3.1.3 Recovery and Grow

After G_F has been seeded over the G_T. The recovery of G_F has a systematic checking over the secret of the attacker. The first thing is to identify the u over the G_T for the head vertex v_h by the degree comparison. Then the ordered sequence degree $S_D(u)$ for G_T and subsequence of the u's initial seed are checked with the corresponding secrets of the attacker. [u's 1-hop neighbour nodes and u's-2 hop neighbour except 1-hop neighbour nodes are checked with the attacker's corresponding secrets]. If the finger print graphs satisfies these attackers secret checks, then it is identified with G_F and its neighbour are identified with V_S by the comparison of the subsequence secret.

After anonymization of the target graph G_T on which the finger print graph G_F is planted on it, the attacker checks the vertices in the target graph G_T with the secrets of G_F which he held. For example, the attacker checks for the vertices with degree 6 in the G_T. Once the candidate head vertex with degree 6 is identified, the attacker isolate it with its immediate neighbours by considering as the candidate finger print graph. The attacker found the internal degree sequence is same to the V_F. Then again he isolate's v's 2-hop neighbourhood, excluding the 1-hop neighbourhood and check the ordered internal degree of the rest of the nodes which matches the secret again. By doing this attacker confirms that he found G_F in G_T. Until all the social actors in the network are de-anonymized the recovery is done. This is how the *Grow* is made. The motivation of implanting the head vertex in seed construction stage shows no back tracking is needed for identifying the G_F as in the previous studies [2,3].

The complexity of recovery algorithm is $O(N^2/V_T)$.

TABLE 3.2: Summary of seed recovery and grow.

For all node $u \in G_T$
if $\deg(u) = \lvert V_F \rvert - 1$ then
$U \leftarrow$ exact 1-hop neighbourhood of u
for all $v \in U$ do
$d(v) \leftarrow$ number of v's neighbours in $U \cup \{u\}$
end for
$s(u) \leftarrow \mathrm{sort}(d(v) \vert v \in U)$
if $s(u) = S_D$ then
$V \leftarrow$ exact 2-hop neighbourhood of u
for all $w \in V$ do
$U(w) \leftarrow$ w's neighbours in U

```
                    s(w) ← sort(d(v)|v ∈ U(w))
                end for
                if hs(w)|w ∈ V i = hSD(v)|v ∈ VSi then
                    {w ∈ V is identified with v ∈ VS
                if s(w) =SD(v)}
                end if
        end if
                    end if
end for
```

4 PERFORMANCE EVALUATION

The performance evaluation of Seed then Grow algorithm is conducted by simulation of small network. The Social Network Database is collected from the real-world. The database which consists of the friendship of social actors, which has 5.2 million actors and 72 million relationship [26]. The performance of Seed algorithm on this data is to implant attacker and the performance of Recovery is to grow the seed and identifying all the target actors in the database. We derived the target and background graphs from each dataset and used their shared vertices as the ground truth to measure against.

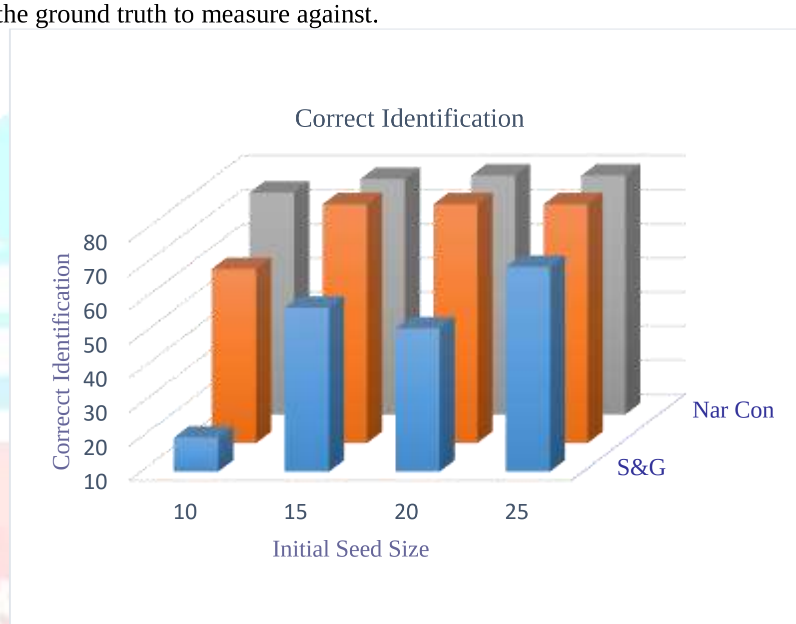

Fig 4a. Graph of Correct Identification of the Social actors on the anonymised social network.

Comparison of Seed then Grow and Narayanan Aggressive algorithm is done [2] Fig 4a and 4b. The Narayanan algorithm performance vary with increase in the threshold accordingly decrease in the accuracy. Here we are varying the threshold in two ways i.e. Narayanan Aggressive and Narayanan Conservative. The aggressive was having ambiguous identification where conservative was having ambiguous identification too. In the proposed algorithm the correct identification of the nodes are more accurate than the previous work. The Unique and asymmetric identification of the actors is done in this proposed algorithm and hence how the accuracy is improved.

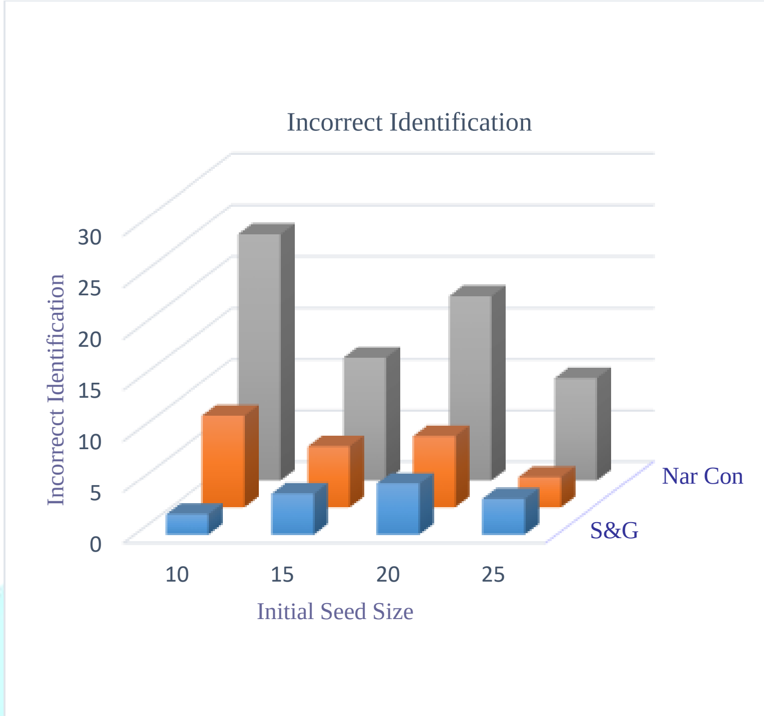

Fig 4b. Graph of Incorrect Identification of the Social actors on the anonymised social network.

4.1 Experiment with respect to time

Initially creation of the social network is done Fig[4.1a]. The nodes are published on the network, at this stage nodes has their identity. Anonymization is done Fig[4.1b], removing the ID of the social actors are done and published on the network. After the anonymization, we are hacking the anonymized social graph and we plant a seed Fig [4.1c], recover it Fig[4.1d] and grow Fig[4.1e]. Finally all the nodes are re-identified Fig[4.1f] [The process is known as de-anonymization]. Hence how only the anonymization of network is not at all sufficient is shown.

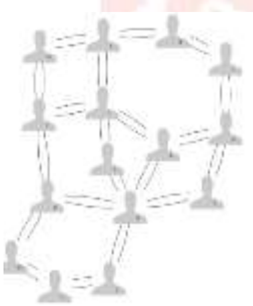

Fig 4.1a. Network of Social Actors

Fig 4.1b. Anonymised social network

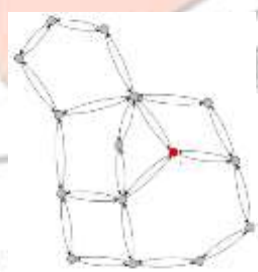

Fig 4.1c. Seed implantation

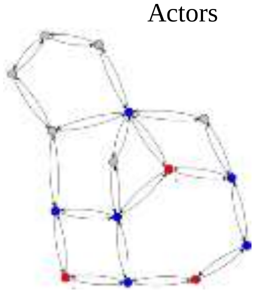

Fig 4.1d. Seed recovery

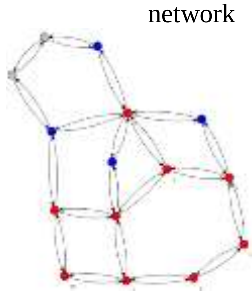

Fig4.1e. Seed recovery and grow

Fig 4.1f. De anonymization of

social networking graph

Seed and grow algorithm tells that once the algorithm has successfully planted the seed, the no of iterations to identify all the actors decreases as the number of seed increases Fig 4.2. The database of the users are maintained in the background, the input for the algorithm is given from the database. The size of seeds could be of attacker's knowledge.

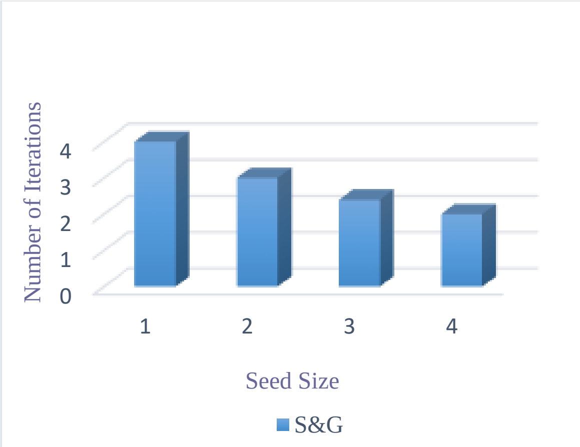

Fig 4.2. Performance Graph With Respect to Iterations

From the Fig 4.2 it is clear that as the No of Seed increases, the No of iterations to identify the actors' decreases.

5. CONCLUSION AND FUTURE WORK.

The proposed algorithm is Seed then grow to re-identify from the social anonymized graph. The design of the algorithm is based on the comparison of the structural similarity of the background graph and the target graph. Algorithm initially identifies the seed graph (star graph) then this is mapped on to the target graph. Seed is made to grow by the structural similarities and the background knowledge of the user which is maintained by the attacker. Nodes are grown this all the social actors are identified in the anonymised social graph. This algorithm eliminates the previous work's ambiguity on identifying seed and also this algorithm is superior in planting many number of seeds and identifying the social actors on the anonymized social graph.

The nodes and the edges after the anonymization can be jumbled and also some can be removed to maintain more privacy against the attacker. Simply to tell after the anonymization the utility of the graph could be changed, which confuse the attacker and keeps the privacy. How much the nodes and edges are jumbled that much it is less useful to the attacker. And also the link encryption can be done after the anonymization of graph.

The defence for this algorithm can be provided by sending a notification for the social actor's personal device as well as to their profiles that you privacy over the network is lost by the user who is holding the ID so and so. Then after the social actor who is notified with this message can remove the relationship (remove edge on the social graph) and is how social actor is defencing over the attacker.

REFERENCES

[1] B. Krishnamurthy and C. E. Wills, "Characterizing privacy in online social networks," in Proc. ACM WOSN, 2008.
[2] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in Proc. IEEE S&P, 2009.
[3] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thour3579x?: anonymized social networks, hidden patterns, and structural steganography," in Proc. ACM WWW, 2007.
[4] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava, "Anonymizing social networks," Univ. Massachusetts, Amherst, Tech. Rep., 2007.
[5] E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationships in graph data," in Proc. ACM SIGKDD, 2007.
[6] A. Korolova, R. Motwani, S. Nabar, and Y. Xu, "Link privacy in social networks," in Proc. ACM CIKM, 2008.
[7] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks," in Proc. Intl. Conf. on Data Engineering (ICDE). IEEE, 2008.
[8] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis, "Resisting structural re-identification in anonymized social networks," VLDB Endowment, vol. 1, no. 1, pp. 102–114, 2008.
[9] J. Scott, Social network analysis: a handbook. SAGE Publications, 2000.
[10] K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Incognito: efficient full-domain k-anonymity," in Proc. ACM ICMD, 2005.
[11] 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, pp. 12–22, 2008.

[12] A. Mislove, H. Koppula, K. Gummadi, P. Druschel, and B. Bhattacharjee, "Growth of the flickr social network," in Proc. WOSN. ACM, 2008.
[13] R. Xiang, J. Neville, and M. Rogati, "Modeling relationship strength in online social networks," in Proc. ACM WWW, 2010.
[14] J. Douceur, "The sybil attack," LNCS, vol. 2429, pp. 251–260, 2002.
[15] N. Tran, B. Min, J. Li, and L. Subramanian, "Sybil-resilient online content voting," in Proc. USENIX NSDI, 2009.
[16] S. Park, B. Aslam, D. Turgut, and C. Zou, "Defense against sybil attack in vehicular ad hoc network based on roadside unit support," in Proc. IEEE MILCOM, 2009.
[17] C. Lesniewski-Laas and M. Kaashoek, "Whanau: A sybil-proof distributed hash table," in Proc. USENIX NSDI, 2010.
[18] C. Chen, X. Wang, W. Han, and B. Zang, "A robust detection of the sybil attack in urban vanets," in Proc. IEEE ICDCS, 2009.
[19] H. Yu, M. Kaminsky, P. Gibbons, and A. Flaxman, "Sybilguard: defending against sybil attacks via social networks," ACM SIG-COMM CCR, vol. 36, no. 4, pp. 267–278, 2006.
[20] H. Yu, P. Gibbons, M. Kaminsky, and F. Xiao, "Sybillimit: A nearoptimal social network defense against sybil attacks," in Proc. IEEE S&P, 2008.
[21] B. Viswanath, A. Post, K. Gummadi, and A. Mislove, "An analysis of social network-based sybil defenses," ACM SIGCOMM CCR, vol. 40, no. 4, pp. 363–374, 2010.
[22] W. Wei, F. Xu, C. Tan, and Q. Li, "SybilDefender: Defend against sybil attacks in large social networks," in Proc. IEEE INFOCOM, 2012.
[23] S. Sorlin and C. Solnon, "Reactive tabu search for measuring graph similarity," LNCS, vol. 3434, pp. 172–182, 2005.
[24] P. Erd¨os and A. R´enyi, "On random graphs," Publicationes Mathematicae, vol. 6, no. 26, pp. 290–297, 1959.
[25] R. Kumar, J. Novak, and A. Tomkins, "Structure and evolution of online social networks," in Proc. ACM SIGKDD, 2006.
[26] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in Proc. ACM IMC, 2007.
[27] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney, "Statistical properties of community structure in large social and information networks," in Proc. ACM WWW, 2008.
[28] M. Porter, J. Onnela, and P. Mucha, "Communities in networks," Notices of the AMS, vol. 56, no. 9, pp. 1082–1097, 2009.
[29] Wei Peng, Student Member, IEEE, {Feng Li, Xukai Zou}, Member, IEEE, and Jie Wu, Fellow, IEEE
"A Two-stage Deanonymization Attack Against Anonymized Social Networks" [29] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins, "Microscopic evolution of social networks," in Proc. ACM SIGKDD, 2008.
[30] A. Barab´asi and R. Albert, "Emergence of scaling in random networks," Science, vol. 286, no. 5439, pp. 509–512, 1999.
[31] D. Soares, C. Tsallis, A. Mariz, and L. Silva, "Preferential attachment growth model and nonextensive statistical mechanics," Europhysics Letters, vol. 70, p. 70, 2005.
[32] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in Proc. IEEE S&P, 2008.
[33] C. Wilson, B. Boe, A. Sala, K. Puttaswamy, and B. Zhao, "User interactions in social networks and their implications," in *Proc.* ACM EuroSys, 2009.