I J C T A, 9(3), 2016, pp. 1461-1472

Detection of Sybil Attack on Social Network Using Sybil Defender Algorithm

Saranya V.*, J.I. Sheeba and Sathya Bama K.*****

Abstract: Peer-to-peer and other decentralized, distributed systems are known to be particularly vulnerable to Sybil attacks in which an adversary creates many bogus identities, called Sybil identities. In Sybil attack, attackers use several identities at a time or it take-off identity of some trustworthy node present in the social network. Since the Sybil identification algorithm provides a solution only in context of peer in a single network. Additionally, there is no technique will be provided to detect sybil attack which is exist from a distributed environment. In order to overcome this problem in a proposed framework, Sybil attack will be detected by using a Sybil defender algorithm. It contains two methods, namely Sybil identification algorithm and Sybil community detection algorithm by using this method Sybil node will be effectively detected.

Keywords: Sybil attack, Sybil identification algorithm, Sybil defender, Social network

1. INTRODUCTION

A social network is a social structure made up of a set of social actors (such as individuals or organizations), sets of dyadic ties, and other social interactions between actors. The social network perspective provides a set of methods for analyzing the structure of whole social entities as well as a variety of theories explaining the patterns observed in these structures. Decentralized distributed systems (such as peer-to-peer systems) are particularly vulnerable to Sybil attacks [1], where a malicious user pretends to have multiple identities (called Sybil identities or Sybil nodes).

A Sybil can delay all the messages by a forward lookup to an incorrect or non-existent peer. Finally, it can send false responses to the receiver. In Sybil attack, an attacker introduces itself in the network with lots of identities, if an attacker gets large network identities it can control a large portion of the network. When an attacker wants to join the network it uses other fake nodes [2].

Sybil attack will create a severe and pervasive problem. For example, it is possible to rig internet polls by using multiple IP address to submit votes to gain advantage in many results of a chain letter [3].

A Sybil attack is also used by companies that increase the Google page rank rating of the customer. Hence defending against Sybil attack is quite a challenging task these attacks have pushed away potential business firms and individual whose get a better benefit from e-commerce application with minimal losses by providing bogus information [3].

In the existing system, it will able to detect Sybil node which is present in a single group or single network using Sybil identification algorithm.Since it does not provide any solution to identify a Sybil node present in a distributed environment [4].

* PG Student, Department of Computer Science and Engineering Pondicherry Engineering College, Puducherry-605014, *Email: sarav0503@pec.edu*

** Assistant Professor, Department of Computer Science and Engineering Pondicherry Engineering College, Puducherry-605014, *Email: sheeba@pec.edu*

*** PG Student, Department of Computer Science and Engineering Pondicherry Engineering College, Puducherry-605014, *Email: sathii_manju@pec.edu*

So in order to overcome this problem in a proposed framework is going to apply Sybil defender algorithm to detect Sybil node which exists from distributed environment in a social network [5].

The main contributions of this work includes: Based on performing a limited number of random walks within the social graphs, this proposed Sybil identification and sybil community detection algorithms are more efficient than previous techniques for large social networks.

The rest of the paper is organized as follows: section2 describes related works, section 3 describes the proposed framework, section 4 describes experimental results and discussion and section 5 describes conclusion.

2. RELATED WORKS

One promising way to defend against Sybil attacks in social networks is to leverage the social network topologies.

Ankush et al., [2] proposed a parental control algorithm to detect a Sybil attack in distributed peer to peer network based on the reputation scheme since this algorithm applicable only for static network.

Guojun Wang [4] have proposed a Sybil identification algorithm this algorithm group all the peers which having similar behavior it act like an identifier source.

Haifeng Yu [5] have provided a Sybil guard approach in order to detect a Sybil peer present in a social network by generating a random path using pre-computed permutation.

H. Yu, P. Gibbons, M. Kaminsky, and F. Xiao [6] have proposed a Sybil limit in this method number of Sybil nodes accepted is reduced by a factor of minus (radicn).

N. Tran et al., [7] have proposed a Gatekeeper is another decentralized Sybil defense scheme that heavily relies on the assumption that the social networks are random expander. This is a strong assumption that has not been validated by previous research. This evaluation shows that Gatekeepers suffers from high false positive and negative rates and cannot effectively identify Sybil nodes on the real-world asymmetric social topologies.

G. Danezi et al., have Sybil Infer, a centralized sybil defense algorithm, leverages a Bayesian inference approach that assigns a sybil probability, indicating the degree of certainty, to each node in the network. It achieves low false negatives at the cost of high computation overhead [8].

A.Kurve and G. Kesidis [9] proposed Sybil detection via distributed sparse cut in this method; it identifies attack edges and quarantine Sybil clusters. This method works well with dynamic trust graphs as nodes do not need to store any pre-computed data.

C.Hota [10] have proposed a safeguard algorithm here some arbitrary verifiers are chosen. Each verifier verifies a group of arbitrary nodes, called as suspicious group, by finding paths to each suspicious node and the connection of paths is taken. After connection, the nodes remaining are more likely to be Sybil.

G. Kesidis [11] have proposed a Sybil-proof referral system, which is based on multiplicative reputation chains. Using a multiplicative reputation chain, single step and multi-step referrals can made Sybil proof.

Douceur [12] has proven that without the use of central authority, it is not possible for a system to fully defend against Sybil attack. Hence, in the p2p network, which is fully distributed, Sybil nodes cannot be removed completely from the network.

Samidha et al., [13] proposed Sybil attack detection on p2p network based on enhanced Sybil-resilient protocol.

Above all techniques it mainly focuses on the Sybil attack which is taking place in single network. It does not provide any solution how to detect attacks which is occurring in a distributed environment. In order to overcome this sybil defender algorithm is deployed to prevent Sybil attacks in a social network.

3. PROPOSED FRAMEWORK

The Main objective of the proposed work is to detect Sybil attack which exists from the distributed environment in a social network. In the proposed framework, is going to apply a Sybil defender algorithm in order to detect a Sybil attack. In Sybil defender Design consists of three components: A Sybil identification algorithm, a Sybil community detection algorithm, and two supporting approaches which limiting the number of attack edges. The three components can be used in conjunction to best mitigate Sybil attacks.The main task of the Sybil identification algorithm to determine whether a suspect node is Sybil or not in the social network. Then, it shows how to efficiently detect the Sybil community around a Sybil node using Sybil community detection algorithm

The reason why it needs a second algorithm is that simply supervising all the nodes in the social graph to find the Sybil community is impractical. Finally, both algorithms are built upon the deduction that the number of attack edges is limited. It is shown in figure1.In this task it includes two tasks, namely[14]

1. Detection of attack
2. Sybil defender
   a. Sybil identification algorithm
   b. Sybil community detection algorithm

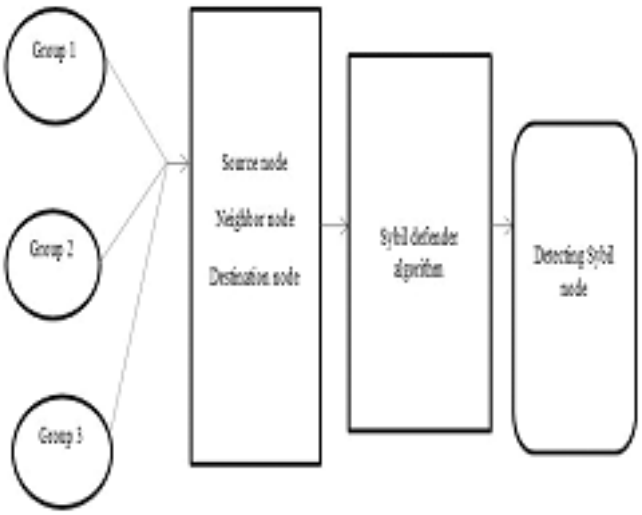

Figure 1: Proposed framework for preventing sybil attack in a social network

1. Detection of attack

In this attack is going to identify the type of attack which is present in the network. Naturally attacks can be classified into two types:

Active attack. It will do all modification activities after listening all incoming and outcomming messages.

Passive attacks: It simply listens to all incoming and outcomming messages, i.e., eaves-dropping, but doesn't harm the system. A peer can be in passive mode and later in active model.

2. Sybil defender

In the Sybil defender algorithm, it is a combination of both Sybil identification algorithm and Sybil community detection algorithm [15]. The working principles of sybil defender algorithm explains briefly in figure 2.

1. Authentication

Login

Input: Provide username and password to get permission for access.

Output: Became authenticated person to request and process the request.

Registration

Input: Provides the user personal details.

Output: create the account for the corresponding user.

2. Admin:

View User Details

Input: Provides the particular user name.

Output: It shows all the details about the particular user.

3. Find Sybil Users

Input: Provides all user relationship in the social network.

Output: It shows the Sybil user in the social network.

Detect Sybil Community

Input: Provides the details about the Sybil user and their relations.

Output: It shows the entire Sybil users and their community.

4. User

Create & Update Accounts

Input: User provides their details.

Output: It creates an individual account for registered users and also updates the account.

5. Making Friends Group

Input: User provides the friend request to their known users in the network.

Output: It creates the friends group for the corresponding user.

6. Share Information

Input: Registered user provides some information and their friends group.

Output: Given information will be shared among the particular friends group.

Finally Sybil node will be weep out using sybil defender algorithm.

A. SYBIL IDENTIFICATION ALGORITHM

A Sybil identification algorithm that takes the social graph G (V, E), a known honest node h, and a suspect node u as input, and outputs whether u is Sybil or not. This algorithm is based on random walks. A sequence of moves of a particle between node of G is term as random walk. If the particle is at node i with degree d_i, then the probability that the particle follows the edge (i, j) and moves to a neighbor j is $1/d_i$.

The main idea behind this Sybil identification algorithm is that, as there is a small cut between the honest region and the Sybil region, the random walks originating from a Sybil node tend to get “trapped” into the Sybil region. Also, because it assume that the size of the Sybil region is not comparable to the size of the honest region, the number of nodes traversed by the random walks originating from an honest node

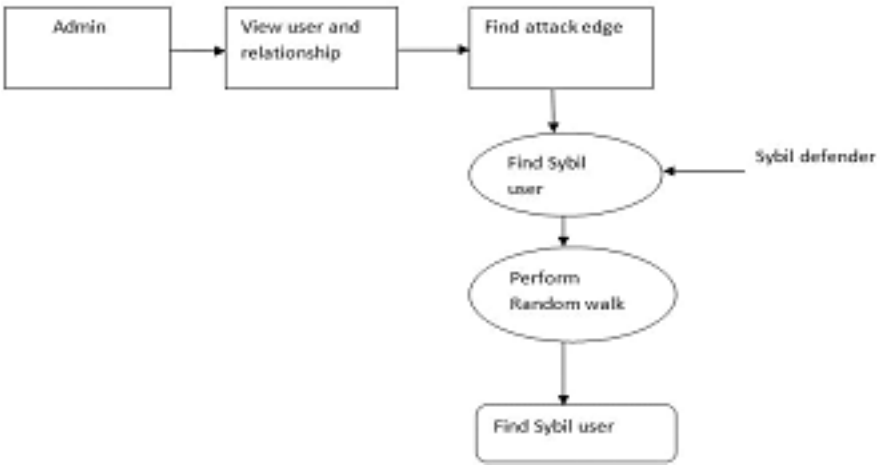

Figure 2: Sybil defender algorithm

will be larger than the number of nodes traversed by the random walks originating from a Sybil node, as long as the random walks are long enough to exhibit the difference between the Sybil region and the honest region, and it performs the random walks many times[14].

The sybil identification algorithm consists of two methods algorithm 1 and algorithm 2.The first methods takes G and h as input and output the thresholds used by second methods to identify sybil nodes. It only needs to be invoked once for social network topology. As shown in algorithm 1, first step 1to 3 performs f short random walks with length ls=log n originating from the node stationary distribution of the honest region. The ending nodes are honest nodes with high probability. After this step 4 to 14 defines the known honest node h and the f ending nodes are treated as judge nodes, from which the algorithm sets up the criterion to identify Sybil nodes. Note the possibility that Sybil nodes may exist in the group of the judge nodes does not influence the effectiveness of the algorithm, due to their very limited number. Starting from a minimum length lmin to a maximum length lmax, with an interval of 100 hops, for each length l, the algorithm performs R (random walks originating from every judge node, and counts the number of nodes whose frequency is no smaller than a threshold t, which is a small constant. The algorithm collects f +1 such values for each length l.Then it computes the mean and

Standard deviation of the f +1 values and outputs a tuple hl; mean; (stdDeviation).

Algorithm 1: preprocessing (G, h)

Step 1: j= {h}

Step 2: For i=1 to f do

Step 3: Perform a random walk with length ls=log n originating from h

Step 4: J=JU {the ending node of the random walk}

Step 5: End for

Step 6: l=lmin

Step 7: While l<=lmax do

Step 8: For i=J.first () to J.last () do

Step 9: Perform R random walks with length l originating from node i

Step 10: Get n_i as the number of node with frequency no smaller than t

Step 11: end for

Step 12: Output (l, mean ({n_i: i€J}), stdDeviation ({n_i: i€J}))

Step 13: l=l+100

Step 14: end while

As shown in Algorithm 2, step 1 to 5 determine whether a suspect node u is sybil, the algorithm first performs R random walks with an initial length $l=l_0$ originating from u. l_0 is larger than or equal to lmin used in Algorithm 1.The algorithm then compares the number of nodes whose frequency is no smaller than t with the mean value in tuple (l, mean, stdDeviation) outputted by Algorithm 1. In step 6 former is smaller than the latter by an amount larger than stdDeviation *α (α=20 in our evaluation), in step 7 to 12 we consider u is Sybil and end the algorithm. Otherwise, the algorithm doubles l and repeats the process, until l is larger than lmax. If u is still not identified as Sybil when the value of l reaches lmax, we consider it honest and end the algorithm [14].

Algorithm2: sybilldentification (G, u, tuples from alg.1)

step1. L=l0

step2. While l <=lmax do

step3. Perform R random walks with length l originating from u

step4. M=the number of nodes whose frequency is no smaller than t

step5. Let the tuple corresponding to length l in the outputs of algorithm 1 be (l,mean,stdDeviation)

step6. If mean−m> stdDeviation* α then

step7. Output u is Sybil

step8. End the algorithm

step9. End if

step10. L=l*2

step11. End while

step12. Output u is honest

B. SYBIL COMMUNITY DETECTION ALGORITHM

After one Sybil node is identified, The Sybil community detection algorithm can be used to detect the Sybil community surrounding it. The Sybil community detection algorithm takes the social graph G (V, E) and a known Sybil node as input, and outputs the Sybil community around us. The Sybil nodes can be identified by using Sybil identification algorithm or any previous scheme. It defines a Sybil community as a subgroup of G consisting of only Sybil nodes, and there is no small cut in this sub graph. The reason it makes this definition is that if a small cut does divide the Sybil region into two parts S1 and S2, and the known Sybil nodes is s in S1, then, from the point of view of us, the honest region and S2 are similar, because there is already a small cut between S1 and the honest region and also a small cut between S1 and S2. When there is a small cut in the Sybil region, this algorithm can detect the Sybil community s.

This algorithm based on performing partial random walks originating from s. Each partial random walk behaves the same as the simple random walks used in the sybil identification algorithm, except that it does not traverse the same node more than once. Therefore, when a partial random walk reaches a node with all the neighbors traversed by itself, this partial random walk is “dead” and cannot proceed. This property makes a partial random walk originating from a Sybil node less likely to leave the Sybil region, compared with a simple random walk, because many such walks “die” when they hit the border of the Sybil region. Similar to the Sybil identification algorithm, the intuition behind this algorithm is that the partial random walks originating from a Sybil node tend to be trapped within the Sybil region, and thus, it can detect the Sybil community by examining the nodes traversed by the partial random walks [14].

The task of Algorithm 3 is to estimate the needed length of the partial random walks. In step 1 to step 8 starting from an initial length l0, the algorithm performs R partial random walks originating from s and

```
Algorithm3. WalkLengthEstimation (G, s)
Step 1: L=l0/2
Step 2: DeadWalkRatio=0
Step 3: While deadWalkRatio<β do
Step 4: L=l*2
Step 5: Deadwalknum=0
Step 6: for i=1 to R do
Step 7: Perform a partial random walk originating from s
        With length l
Step 8: If the partial random walk is dead before it
        Reaches l hops then
Step 9: Deadwalk num++
Step 10: end if
Step 11: end for
Step 12: Deadwalkratio=deadwalknum/R
Step 13: end while
Step 14: Output l
```

counts the ratio of dead walks, which are the walks that cannot proceed before they reach the required length. If this ratio is smaller than β, a threshold close to 1, the algorithm doubles the current length and performs the partial random walks again. This process is repeated until the dead walk ratio is no smaller than β. Then, in step 9 to 14 the algorithm outputs the current random walk length l. The reasoning is that the number of untraversed sybil nodes is very small when the dead walk ratio is close to 1 and with a relatively large R [14].

4. RESULTS AND DISCUSSION

Dataset

The existing system is validated using face book dataset here and also the proposed system is going to evaluate the input from face book dataset.

Performance Metrics

The evaluation metrics are mainly involved to calculate the effectiveness of the performance, the performance of the existing framework is measured in terms of the quality measures namely

1. Non-trustworthy rate
2. Detection rate
3. Packet Delivery Ratio (PDR)
4. End-To-End Delay

Non-trustworthy rate

Non-trustworthy rate is the ratio of the number of honest peers which are erroneously marked as a Sybil peer to the number of total honest peers.

$$Non-trustworthy = \frac{No.\ of\ \ honest\ peer\ marked\ as\ Sybil}{No.\ of\ \ total\ \ honest\ \ peer\ \ rate}$$

Detection rate

Detection rate is the proposition of detected Sybil/malicious peers to the total Sybil/malicious peers.

$$Detection\ Rate = \frac{Detected\ Sybil\ peer}{Total\ Sybil\ /\ malicious\ peer}$$

Packet Delivery Ratio (PDR)

The ratio of the number of data packets receives to the packet data send to the destination. This illustrates the level of delivered data to the destination.

$$PDR = \frac{\sum Number\ of\ packets\ receive}{\sum Number\ of\ packets\ sends}$$

End-To-End Delay

The average time it takes a data packet to reach the destination. This includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue. This metric is calculated by subtracting time at which first packet was transmitted from source from time at which first data packet arrived to destination.

$$End\ to\ end\ delay = \frac{\sum(arrive\ time - send\ time)}{\sum Number\ of\ connection}$$

Existing framework

In existing framework, Sybil node present in a single network gets detected by using Sybil identification algorithm. In the approach group all the node which has similar behavior using similarity trust relationship hence it will act like a identifier source. It can send Identifiers to others as the system regulates. If a node sends less or more, the system can be having a Sybil attack node. The performance of existing system is measured using the above metrics, and also evaluated the inputs from face book dataset.

Experimental results

The below table 1 shows a non-trustworthy rate detection and here number of inputs will be a node which is deployed. The table also shows that when number of honest peer get marked as sybil increases than non-trustworthy also get increases.

The results were shown in the Table 1.This table shows the performance of the non-trustworthy rate detection, which is detected using Sybil identification algorithm.

Table 1
Table For Non Trustworthy Rate Using Different Nodes As Inputs

Technique used	*No of nodes*	*No of honest peer*	*Non-trustworthy rate*
Sybil	20	12	0.5432
identification	30	24	0.5231
algorithm	40	31	0.4389
	50	38	0.4290
	60	52	0.4110

The below graph figure 3 shows a non-trustworthy rate detection, which is detected using Sybil identification algorithm with a number of nodes. Here x-axis denotes number of nodes deployed and y-axis denotes non-trustworthy rate detection.

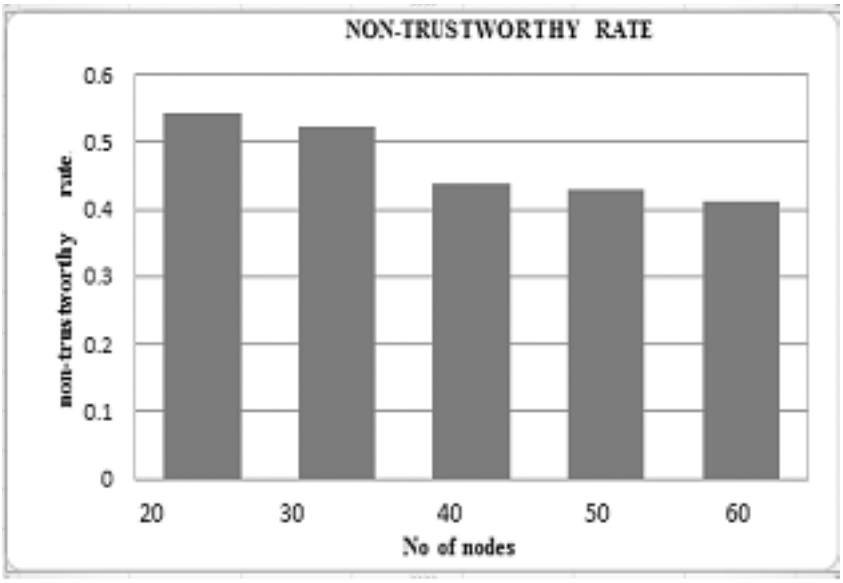

Figure 3: Graph for non-trustworthy rate using different nodes as inputs

The below table 2 shows a detection rate and here number of inputs will be a node which is deployed.

The results were shown in the Table 2. This table shows the performance of the detection rate which is detected using Sybil identification algorithm

Table 2
For Detection Rate

Technique used	*No of nodes*	*Total sybil peer*	*Undetected sybil peer*	*Detection rate*
Sybil	20	12	4	0.639
identification	30	15	8	0.730
algorithm	40	26	15	0.789
	50	35	21	0.850
	60	46	28	0.871

The below graph figure 4 shows a packet delivery detection, which is detected using Sybil identification algorithm with a number of nodes as inputs. Here x-axis denotes number of nodes deployed and y-axis denotes packet delivery detection

The results were shown in the Table 3. This table shows the performance of the packet delivery which is detected using Sybil identification algorithm.

The above graph figure 5 shows a packet delivery rate detection which is detected using sybil identification algorithm with a number of nodes. Here x-axis denotes number of node deployed and y-axis denotes packet delivery.

The results were shown in the Table 4. This table shows the performance of the end to end rate detection which is detected using Sybil identification algorithm.

The above graph figure 6 shows a end to end delivery rate detection which is detected using sybil identification algorithm with a number of nodes s. Here x-axis denotes number of node deployed and y-axis denotes end to end delay.

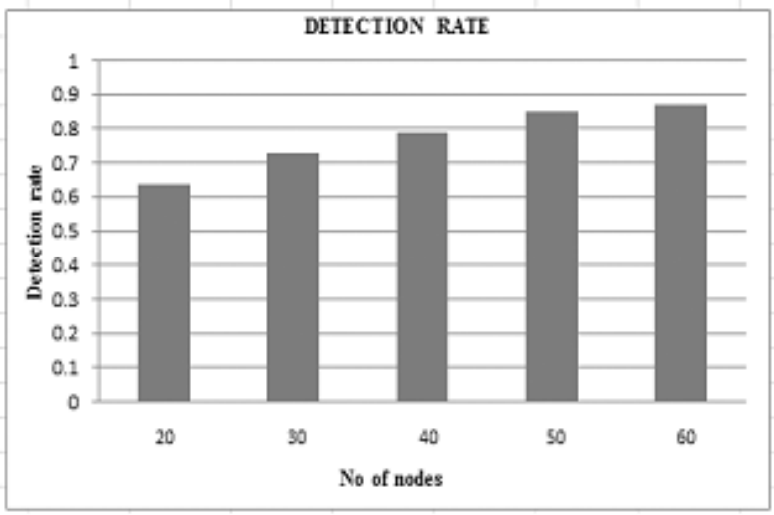

Figure 4: Graph for detection rate using different nodes as inputs

Table 3
For Packet Delivery

No of nodes	*No of packets sends*	*No of packet receive*	*Packet delivery*	*Packet loss*
20	20	15	0.75	5
30	30	30	0.6	10
40	40	40	0.625	15
50	50	50	0.61	20
60	60	60	0.5	15

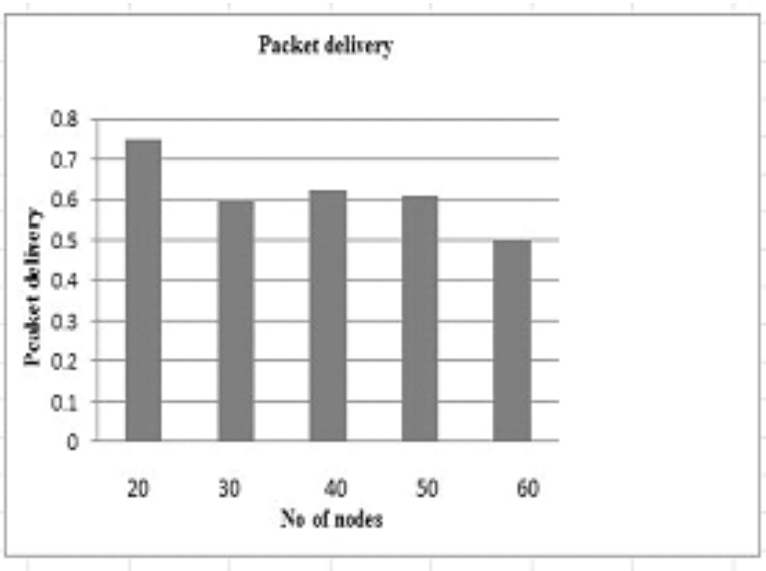

Figure 5: Graph for packet delivery using different node as input

Table 4
For End to End Delay

No of connection	*Send time*	*Arrive time*	*End-to-End delay*
20	10	15	0.25
30	2	4	0.666
40	1	3	0.05
50	2	16	0.28
60	5	6	0.0161

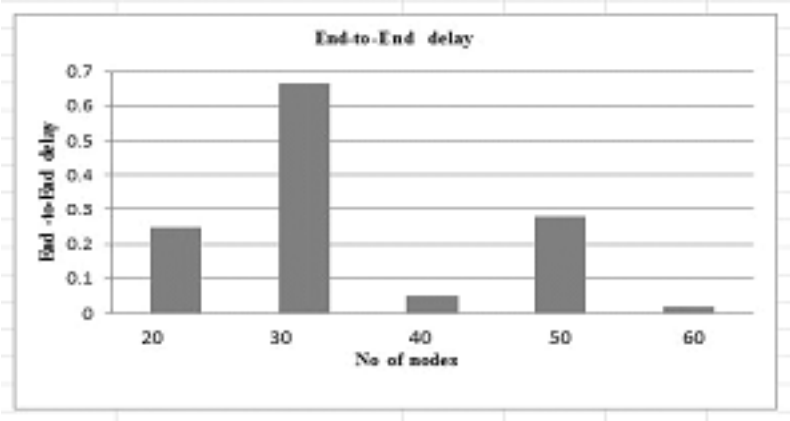

Figure 6: Graph for end to end delay using different nodes as inputs

5. CONCLUSION

Social network plays an ever increasingly important part of daily life. Since it is vulnerable to security attack, namely Sybil attack The Sybil attack will be detected in a social network by using Sybil defender algorithm. This will produce good result and reduce the time complexity compared to a previous Sybil identification algorithm.

Acknowledgment

I express the sincere thanks to our Institution for extending the infrastructural facilities to carry our work successful

References

[1] A. Cheng, and E. Friedman, "Sybilproof reputation mechanisms", In Proc. ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems", 2005, ACM Press, pp. 128-132.

[2] Ankush Tehale "Parental control algorithm for sybil detection in distributed p2 p networks" in International Journal of Scientific and Research Publications, Volume 2, Issue 5, May 2012.

[3] Jochen Dinger and Hannes Hartenstein, "Defending the Sybil Attack in P2P Networks: Taxonomy, Challenges, and a Proposal for Self Registration", In Proc. First International Conference on Availability, Reliability and Security (ARES 2006), Vienna, Austria, 2006, pp. 756-763.

[4] Guojun Wang "Neighbor Similarity Trust against Sybil Attack in P2P E-Commerce" in IEEE transactions on parallel and distributed systems, vol. 26, no. 3, March 2015.

[5] Haifeng Yu, M. Kaminsky, P. B. Gibbons, A. Flaxman, "SybilGuard: Defending Against Sybil Attacks via Social Networks", in Proc. ACM SIGCOMM, 2006, pp. 267–278.

[6] Haifeng Yu, Phillip B. Gibbons, M. Kaminsky, F. Xiao, "SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks", in Proc. IEEE/ACM transactions on networking, vol. 18, no. 3, June 2010.

[7] N. Tran, J. Li, L. Subramanian, and S.S. Chow, "Optimal Sybil Resilient Node Admission Control," Proc. IEEE infocom, 2011.

[8] G. Danezis and P. Mit, "Sybil infer: Detecting Sybil Nodes Using Social Networks," Proc. Network and Distributed System Security Symp. (NDSS), 2009.

[9] A. Kurve and G. Kesidis, "Sybil Detection via Distributed Sparse Cut Monitoring", in Proc. ICC 2011, pp. 1-6.

[10] C. Hota, J. Lindqvist, K. Karvonen, A. Ylä-Jääski, Mohan C.K.J"Safeguarding Against Sybil Attacks via Social Networks and Multipath Routing", in Proc. NAS 2007, pp. 122 – 132.

[11] G. Kesidis, A. Tangpong and C. Griffin, "A sybil-proof referral system based on multiplicative reputation chains," IEEE comm. letters, pp. 862-864, nov. 2009.

[12] J. Douceur "The Sybil Attack. In 1st International Workshop on Peer-to -peer Systems" (IPTPS '02). Springer, 2002, pp. 251-260.

[13] Samidha Nagdeve “Sybil attack detection on peer to peer network based on enhanced Sybil –resilient protocol” in international journal for scientific and research and development.

[14] Wei Wei, Fengyuan Xu Sybil Defender: “A Defense Mechanism for Sybil Attacks in Large Social Networks” in IEEE transactions on parallel and distributed systems, vol. 24, no. 12, December 2013.