# Sybil Resilient Identity Distribution in P2P Networks

Jissy Liz Jose
Department of CSE
SCT College of Engineering
Trivandrum, Kerala, India
jissyliz@gmail.com

Jayasudha JS
Department of CSE
SCT College of Engineering
Trivandum, Kerala, India

Sabu M. Thampi
School of Computer Science
Indian Institute of Information
Technology and Management -Kerala,
India

## ABSTRACT

Sybil attack is one of the most challenging problems related to identity management in Peer-to-Peer networks. The huge number of fake identities created by malicious users may attempt to gain a large influence on the network and may collude or subvert the system. This paper proposes a Sybil defense for super peer based P2P networks where identities are distributed in a hierarchical way from super nodes to peers. The identification scheme is based on invitations and assigning a set of identities to each node is based on how they utilized the earlier ones and referral values.

## Categories and Subject Descriptors

C.2.0 **[Computer-Communication Networks]:** General-*Security and protection*

## General Terms

Management, Security

## Keywords

P2P networks, Sybil attack, Sybil defense, Sybil identities

## 1. INTRODUCTION

A Peer to Peer (P2P) network is a distributed network composed of a large number of distributed, heterogeneous, and independent peers. P2P networks provide an alternative to the traditional client-server communication model in which a node in a P2P network can act as a server and a client at the same time. The P2P computing provides properties like no central point of failure and no service bottlenecks by decentralizing the service among participating nodes. In recent years many research work have been done and are still in progress to improve their robustness, security and scalability. P2P networks are less secure than a client-server network because of their decentralized nature. P2P specific security problems include [1] targeted denial of service attacks, forgery, pollution attack, sybil attack, attacks on routing queries and attacks on data integrity.

For efficient routing and load balancing in P2P networks, every node should have a unique identifier and there should be an identity management scheme for handling identities in distributed environment. The term ‘identity’ refers to information about an entity that is sufficient to identify that entity in a specific context and any entity, either in the digital world or in real world, is associated with an identity. Identity management plays a crucial role in improving a network’s operational efficiency, management control and cost savings. Systems need to manage the growing number of users, their dynamicity, their access to information and applications scattered across heterogeneous systems.

Identity management includes three major aspects: *acquisition of identities, authentication,* and *authorization*. Authentication is the process that verifies the association between an entity and the corresponding identity. Different authentication mechanisms have been widely used like password checking, challenge and response and biometric verification. Authorization grants the permission to an authenticated identity for accessing resources that it is eligible for (example, by using Access Control List).

This paper focuses on the issue of acquisition of identities by entities. In P2P networks, the issues related to identity management may fall in the areas [1] like secure assignment of node identities, entity-identity association, distributed trust among peers, resistance among malicious peer’s collusion, robustness and damage discovery. Distributed environment works on the usual assumption that each participating entity controls exactly one identity. When this assumption is non-verifiable, the system is subject to a condition in which an individual entity generates huge number of fake identities. This problem, named as *sybil attack* [3] refers to a situation in which individual malicious users may join the network multiple times under multiple fake identities and these fraudulent ones may try to inflate their own reputation in the network to appear more trustworthy than they really are and the integrity or the availability of the P2P network may be disrupted. This attack usually happens when obtaining a new identity is not expensive. Sybil identities can easily overcome the genuine users in various collaborative tasks, recommendation systems, redundant/false routing and data replication in DHTs.

Sybil defenses aim at limiting the number of sybil identities, but false positives and false negatives [4] are acceptable to an extent in this process. Complete elimination of false negatives (assigning sybil identity as genuine) are not necessary since the distributed system should be able to tolerate some fraction of byzantine identities, otherwise, even without sybil attack it is not

*ICACCI '12,* August 03 - 05 2012, CHENNAI, India
Copyright 2012 ACM 978-1-4503-1196-0/12/08…$10.00.

robust. Thus, a sybil defense should be able to limit the total number of false negatives below the tolerance threshold of the system. Most of the applications can easily tolerate with a small fraction of false positives (labeling genuine nodes as sybil). For example, in a P2P backup system, if a node considers another one as sybil, then it will not trust that node for storing its data. A false positive rate of 20% means that, the given node will still trust 80% of the genuine nodes, and can use these. Also, if it is a recommendation system, then it can use votes from 80% of the genuine identities. From these observations, we can conclude that defenses against sybil attack permits some bounded fraction of false positives and false negatives also.

In this paper, we propose a sybil defense based on invitations and systematic distribution of identities. Initially, each peer assigns a set of identities to peers invited by them and later based on how they utilized the earlier ones. As the network grows, the super nodes occasionally computes the rank matrix based on the transaction between peers and considers these referral values also, for assignment of new set identities when nodes request more.

The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 discusses the proposed method for assignment of identities, distribution of identities, and calculation of rank matrix. Section 4 describes the simulation and analysis and finally, Section 5 concludes the paper.

## 2. RELATED WORK

In the absence of a centralized authority concurrently certifying all identities, a possibility of sybil attack always exists and it was first proposed by J. Douceur in [3]. Many papers suggest certification [5, 6] as a solution to the sybil attack, and it is the most common solution. But trusted certification usually depends upon a centralized entity which ensures that each entity is associated with exactly one identity.

Resource testing [7] is another approach to defend sybil attack in which it tries to check whether a number of identities possess fewer resources than would be expected if they were independent. These checks include tests for processing power, memory capacity, network bandwidth etc. This can be considered as a minimal sybil defense, but for many applications it is not sufficient if an attacker is able to obtain large number of identities for a successful attack, even if it is expensive. Another approach is to impose a fees [8] (one time cost) for obtaining an identity. However, here the issue is how to put a limit on fees such that it must be low enough to allow everyone to join, but also be high enough to prevent malicious users from obtaining many identities.

Recently, many mechanisms leverage social networks [9, 10] for limiting sybil attack. A social network refers to an undirected graph where vertices of the graph correspond to nodes/identities and edges correspond to human established trust relations between users. Social network based schemes work on the assumption that, even if a malicious user creates a large number of sybil identities, it can establish only very few edges with genuine nodes. So, the network can be divided into sybil region and non-sybil region by computing the minimum cut along the graph. The main drawback is that these techniques depend on the limited availability of real-world friendship edges between nodes and P2P application in use may have only little intersection with this. Similarly, these friendship relationships are difficult to construct as it requires out of band communication.

There are many other sybil defenses which do not leverage social networks such as *sybil defense mechanism based on network coordinates* [12], in which the scheme offers guarantee under certain assumptions on the network position of the attacker. Dsybil [11] uses user feedbacks to defend against sybil attacks in recommendation systems. Another approach is a referral system based on multiplicative reputation chains [14] in which it shows how a reputation system with chain referrals adds referrals from different referral paths/chains, is sybil-proof. In [13] an existing member has to invite another user for obtaining an identity in the network. This method is based on the construction of a perfect tree representing social relationship between users. The top of the tree consists of founding members which emulate root node by sharing private key of the root through threshold cryptography. The invitations are delivered based on the value of a factor parameter which is calculated by each node based on their local policies. Here the term ‘weight’ of a node corresponds to number of invited nodes and ‘potential’ corresponds to maximum attainable weight. The algorithm tries to construct a perfect tree in which, for each child, weight is equal to the potential and for each parent their children have a potential corresponding to the factor parameter. The performance depends on proper selection of value of the factor parameter.

Our proposed scheme is also based on invitations and is suitable for P2P service providers for admission control. We propose a super peer based P2P networks where identities are distributed in a hierarchical way from super nodes to peers. Here assigning a set of identities to each peer (for inviting others) is based on how they have utilized the earlier ones. Initial assignment of identities is based on the value of a parameter ‘k’ and on further requests; a timestamp based calculation is done by the parent node (see Section 3.1). Meanwhile peers contact each other, and one peer rates the other one on the basis of service they received. A super peer computes referral values of rank matrix.

## 3. THE PROPOSED IDENTITY DISTRIBUTION SCHEME

The proposed work is intended to prevent a node from obtaining a huge number of fake identities. This scheme is based on invitations and distribution of a set of identities (which can be used by a node for inviting others) to each node in the network. For obtaining an identity on the network or, for a node to become a part of the network, it has to be invited by an existing member [13]. When a node joins the network, a set of identities are assigned to each node by the parent node for inviting others. So there should be a control on the count of identities offered to each node, otherwise a malicious node could create an unlimited number of sybil identities either directly by inviting them or indirectly by inviting some sybils which in turn invite other sybils. The proposed scheme intends to prevent an attacker from creating an unlimited number of sybil nodes, even though a genuine node can invite new nodes. Thus, the growth of the network is on the basis of how identities are assigned to each node and how they use it.

The proposed scheme can be used by P2P service providers for node admission, by limiting the entry of sybil identities into the system. Initially few pre-trusted peers with sufficient CPU, memory and network bandwidth are assigned as super nodes by the service provider. So we consider a P2P network where nodes can be categorized into two: peers/regular clients and super nodes/super peers. The super nodes and peers are interconnected. A group of peers will be monitored by a super node and every super node is connected to at least another super node. The network topology is shown in Figure 1. For a node to join the network, it must be invited by some member (it can be a super node or peer) in the network. Every super peer is assumed to have a set of invitations/identities (say *N*). When a node invites another node, the former is called parent and the latter is called the child.

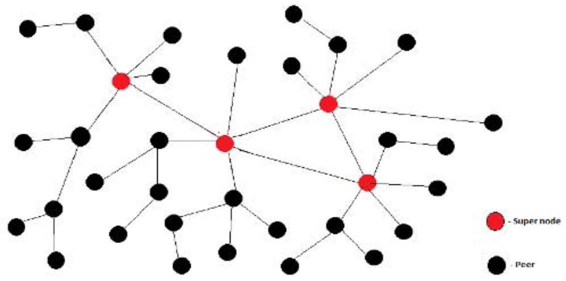

**Figure 1. Network Structure**

When a node accepts the invitation from another node, the parent assigns a unique identifier and a set of identities (for inviting others) to its child. Here the issue is what fraction of identities from parent is to be assigned to the child (see Section 3.1). Each node (either super node or peer) in the network is assumed to have the following parameters:

- A unique identifier
- Identifier of parent
- Identifier of the super node under which it comes
- A public key- private key pair
- Count of used Invitation
- Range of invitations it can use, indicating lower limit and upper limit of node numbers (here invitations correspond to set of identities.)
- A Timestamp assigned to each child by the parent when a set of identities are assigned to it.

The first three parameters are identical in case of a super node (identifier of the super node). The unique identifier of a peer is composed of two parts. The first part indicating the super node under which it comes and second part is the node number under the corresponding super node. Each node will store the unique identifiers of nodes it has directly invited, along with their assigned range of invitations and timestamp showing when it has assigned those invitations. In addition to these, if the node is a super node it will maintain a list of all nodes under it, either directly or indirectly invited.

## 3.1 Dispatch of Identities

For those nodes which are directly invited by super nodes (at level 1) the super node assigns a set of identities to them for inviting others. Now these nodes can invite other nodes and they too can invite other nodes and this procedure goes on. For inviting other nodes into the network, existing nodes in the network should have unused set of identities among the set of invitations granted from their parent. Generally parent node (super node or peer) will give only small fraction of invitations to its child from its available set of unused invitations, when a child accepts its invitation. Here we limit the invitations count by giving only $(k/(1+\log_2 k))$th fraction if the node was assigned *k* invitations (set of identities) by its parent. The assignment of identities should be limited as each level passes in the hierarchical structure. In this scheme the super node will assign only around *N/10* identities for nodes at level 1. So even if a node at level 1 is sybil the number of sybils would be below 10%.

When a node has no more invitations to deliver, the node requests its parent. If its parent has unused invitations, then the parent gives invitation from that, to its child. If the parent does not have unused invitations, the node asks its parent, and it continues until it reaches the super node. Thus, each node in the network will have a unique path to the super node and can be found out by tracing the corresponding parent at each level. The super node will consider referral values of the rank matrix also, for identity distribution if it ever done that. When and how this rank matrix is calculated is discussed in section 3.2

The dispatch of invitations by a parent to its child should be in such a way that the growth of the network must be balanced. Suppose a node *n* is an existing member and it has unused invitations and it is inviting another node *c*. Each node is assumed to have the set of parameters mentioned in Section 3. Then the process is described as follows:

Let *k* - Count of identities/invitations assigned to a node
*Parent_id* - Unique identifier of parent
*Supernode_id* - Unique identifier of supernode
*Invit_count* - Count of invitations at any time for a node
*Count_usedinvit* - Count of used Invitations (count of directly invited nodes and set of identities assigned to them)
*LL* - Lower Limit of range of invitations of node
*UL* - Upper Limit of range of invitations of node
*Node_number* - Second part of unique node identifier followed by Supernode_id

**Procedure 1. Invitation of another node when parent has unused set of invitations.**

1. Node *n* invites another node *c*.
   Let *n.k* - count of invitations assigned to *n* initially.
   *n.LL* - Lower Limit of range of invitations of *n*
   *n.UL* - Upper Limit of range of invitations of *n*
2. Node *c* accepts invitation and parent *n* updates parameters of child *c* as:
   *c.Node_number = n.LL*
   *c.Parent_id = n.Unique_id*
   *c.Supernode_id = n.Supernode_id*
   *c.Invit_count*=Integer(*x*)
   $\text{where } x = n.k/(1+\log_2 n.k))$
   *c.Count_usedinvit = 0*
   *c.Timestamp = <Current Time>*
   *c.LL = c.Node_number + 1*
   *c.UL= c.Node_number + Invit_count*
3. The unique identifier of node *c* is *c.Supernode_id* followed by *c.Node_number* and is added to the list of identifiers maintained by the super node
4. The new identity of *c* is added to the list of children of the node *n* along with the timestamp and count of invitations assigned.
5. Parent node *n* updates its parameters:
   *n.Invit_count = n. Invit_count - c.Invit_count +1*
   *n.Count_usedinvit = n.Count_usedinvit + c.Invit_count+1*
   *n.LL = n.LL + Invit_count*

The growth of network under one super node (named as sp1) is shown in Figure 2. We assume *N*=500, (initial count of invitations of super node). When it invites another node the child's unique id is sp1_nd1, along with value of k as 55 (initial count of invitations of sp1_nd1) and range of invitations as 2-56. When sp1_nd1 invites another node, its id will be sp1_nd2 with value of k as 9 and range of invitations as 3-11. Then its parent, sp1_nd1 will have count of invitations 45 and range of invitations 12-56. When super node itself calls another node, its id will be sp1_nd57 with value of k as 55 and range of invitations as 58-112. As each level passes, k value decrements as 500, 55, 9, 2, 1 if k value of super node is 500 *(N=500)*. If the k value of super node is 1000, then k value decrements as 1000,100,21,4,1 or if k value of super node is 2000, then k value decrements as 2000,181,22,1. From these it is clear that the distribution of identities is limited as each level passes, even though the initial count of identities is high.

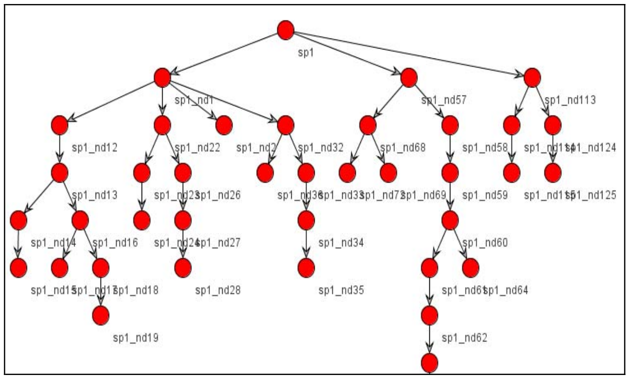

**Figure 2. Growth of network under one super node (sp1)**

Normally, a malicious node tries to invite more nodes than its siblings, so that the parent has to assign invitations to its children in a balanced way. A malicious node may use all its invitations instantly for obtaining a large part of identifiers, but a genuine node will use his invitations over time. So the despatch of invitations must be gradual. Therefore a monitoring on the usage of invitations is essential and hence we associate a timestamp with every node when it is granted a set of identifiers by its parent. The usage of invitations is compared with their siblings to find out whether they invite notably more nodes than its siblings. Each node can invite others until count of invitations reaches 0 or less than *k* value of the child. If a node gets invitations from many members, then it will accept invitation from only one of them. So the count of invitations of a member should be reduced only after a node accepts its invitation.

Suppose a node *n* wants to invite another node *c* and node *n* has no unused invitations, then the process of node admission or invitation dispatch and distribution of identities is described as follows: (variables represents same as in procedure1)

**Procedure 2. Invitation of another node when parent has no unused set of invitations.**

```
1. Check the count of unused Invitations of node n
   If n.Invit_count < c.k
       Node n is either terminal or any node which runs out of
       invitations and requests its parent node p.
2. Node p compares node n's following parameters with siblings
       a. Count of invitations assigned previously
       b. Duration between requests (comparing Timestamp with
          current system time)
       c. Frequency of usage of invitations compared to general
          behavior of network
       d. Reputation value in total rank matrix if ever super node has
          done the calculation
       And if no peculiarities are observed, then:
           If p.Invit_count > c.k
                   Node p assigns c.k invitations to node c and
                   accordingly update LL, UL and Invit_count of p
                   and n, then
                       follow the steps in Procedure 1 and exit
3.         Else
                   Obtain Identities from higher levels
                   do {     A = p.parent;
                            p = A
                   }while ( A.Invit_count > c.k )
4.     A assigns k invitations to c and accordingly update LL,UL and
       Invit_count of p and n, then
                       follow the steps in Procedure 1 and exit
5.     If any peculiarities in behavior of c is observed by p in Step2
       then p suspects c as "Sybil" and gives only log k invitations or
       deferred for a definite time period (on further requests).
6.     If request for more invitations propagates and reaches super
       node and super node also runs out of invitation or total count of
       nodes under super node reaches N/5 or its multiples
               do      Calculate_rankmartix
7.     Super node either re-initializes the value of N depending on
       network capacity or a new super node is initialized.
8.   After the procedure Calculate_rankmartix , the referral values are
       broadcasted by the super node and these values are also
       considered on further requests for invitations.
9. Else
           follow the steps in Procedure 1 and exit
```

At any time if a node is suspected as sybil by the parent or super node by either rank matrix calculation or by the observation of usage of invitations, on further request for invitations, it will be assigned only (*log k*) invitations. Later if requests continue, or the intervals between requests are very short compared to others, it will be deferred from getting more. Even if a node creates multiple identities over time and if it has no fraudulent intentions, it will not create problems to the system. But if these fake identities have purpose of an attack using sybil identities (For eg. sybil nodes can initiate a DDoS attack), attacker cannot wait for a very long duration and it will somehow tries to get identities as early as possible. Thus, we employed this timestamp based approach to limit sybil attack. At the same time this scheme will not defer any genuine node from entering the network. In an extreme case if a node wants to become a part of the network and has not yet received any invitation, then it can request any one among the existing nodes having unused invitations to invite it by out of band means using his social relationships. So deferring some nodes as sybils by the system will not deny any genuine node from entering the network.

If a super node runs out of invitations and if it has enough computational power to handle more identities, *N* can be reinitialized. Otherwise, an active node with high reputation along with enough computational power, memory capacity and network bandwidth has to be selected as super node by conducting an election process among all super nodes. Also every super node is associated with one or more super node so that one can act as a monitor to other, so that chances of both of them being sybils are very low.

## 3.2 Calculation of Rank Matrix by Super Node

When request for new set of invitations reaches super node, like all other nodes, it also does an analysis of all nodes comes directly under it. If super node runs out of invitation or total number of nodes under super node reaches *N/5* or its multiples (super node is assumed to have a set of invitations/identities, say *N*), it does a reputation calculation of each node under it (directly/indirectly invited) by using a normalized rank matrix [14,15]. When two nodes contact, and if they are under the same super node, they assign a referral/ reputation value on the other. On further transactions with the same node, based on their behavior, referral values may be incremented or decremented by a constant factor. These referral values are encrypted with the public key of super node and send it to the super node or to the parent so that parent will collect the reputation values send by its children and send it collectively. The super node decrypts these values using their private key so that no other node in between can edit it. In case, if no prior contact occurs between a pair of nodes and also in case of self-referrals, a reputation value of zero is assigned.

Let *RM* be an *(n+1)*(n+1)* direct rank matrix [14] where *n* is the number of nodes under the super node. Each entry $RM_{ij}$ denotes the referral/reputation value of node *i* assigned by node *j*. The minimum referral value is 0.0 and maximum referral value is 1.0. Here

$RM_{ij} = 0$ [ If *i=j* or if no contact between nodes *i* and j]

$RM_{ij} = 0 \text{ or very low}$ [ If node *j* is sybil ]

$RM_{ij} = 1 \text{ or very}$ *high* [ If both nodes *i* and *j* are sybils ]

Otherwise a referral value in between 0 and 1 is assigned on the basis of service provided. The sybil nodes are virtual or non-existent. So it cannot supply genuine files to genuine nodes. Hence, they will be assigned a very low or zero referral value by the genuine nodes. Based on the transactions, the reputation value corresponding to the nodes may vary. If nodes upload good files and possess a good resource sharing capability in the network, then they can achieve a high value. If node *i* creates a sybil identity *j,* they would give high referral values (false referral) to each other in order to obtain a high reputation in the system. Hence, this direct matrix cannot be used by super node for assignment of new identities. Also the direct matrix is normalized column wise so that sum of elements in each column will be 1 to reduce large disparities in reputation values. The indirect rank matrix calculation is done to dilute or to reduce the effect of the false referrals. From the direct referral matrix, we can compute the indirect referral matrix where each node is assigned a reputation value considering the values assigned by all others except it. Here we assume indirect referrals of the multiplicative form [14] where reputation value of peer *i* assigned by peer *j* can be calculated by considering values assigned to *i* by *k* and to *k* by *j* taken over every value of *k*. So indirect matrix is calculated as:

$$I_{ij} = \sum_{k \neq j,i} RM_{ik} * RM_{kj} \qquad (1)$$

where $k = 1,2,\ldots..n$

Now we can calculate the total matrix, $F_{ij}$ using the equation

$$F_{ij} = c\, I_{ij} + (1-c)RM_{ij} \qquad (2)$$

for some non-negative constant $c < 1$. The total matrix calculation is done so that both direct and indirect matrices are considered for final rank matrix calculation. The value of *c* is selected such that if we are using a value greater than 0.5 then we are giving more weightage to indirect matrix and vice versa. The referral values calculated by super node ($F_{ij}$) are also considered for assigning invitations to nodes on further requests.

# 4. SIMULATION AND ANALYSIS

This section describes the preliminary simulation setup and analysis of obtained result. The preliminary simulation environment was created using Java with an initial setup of 5 super nodes and each having initial invitation count of 500. The simulation conducted until 500 nodes joined the network. If the node asks for more invitations to its parent, parent node compares the following parameters with its other directly invited children for labeling a node as either "genuine" or "suspected as sybil".

**Table 1. Parameters used by a node on requests for more invitations from lower level.**

| | |
|---|---|
| 1 | Level of the node in the hierarchical structure, considering super node at level 0 |
| 2 | Count of invitations assigned previously |
| 3 | Number of times it has requested compared with its siblings |
| 4 | Duration between requests (comparing Timestamp with current system time)compared with its siblings |
| 5 | Frequency of usage of invitations compared to general behavior of network |
| 6 | Reputation value in total rank matrix if ever super node has done the calculation. |

Fig. 3 depicts the network structure along different levels and it can be visualized in JUNG. The yellow colored nodes represent the super nodes and red ones represent ordinary nodes.

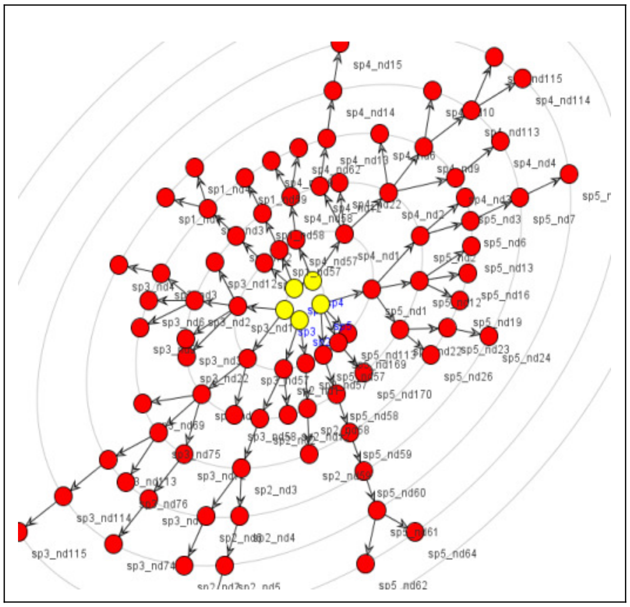

**Figure 3. Network Structure showing different levels**

Fig. 4 shows the growth of network until the total number of nodes reaches 500**.** From this we could infer that even if we put constraints on admission of nodes, the network grows, but it is not uniform.

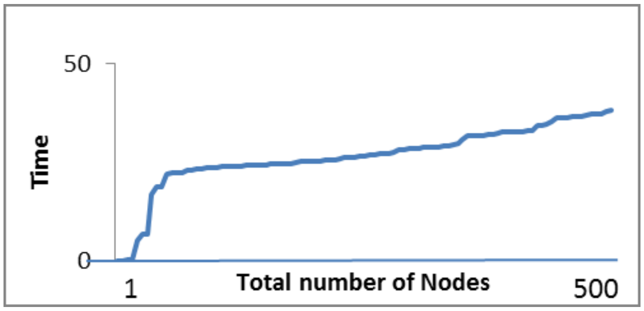

**Figure 4. Node admission rate**

Based on the parameters discussed in Table 1, among the 500 nodes joined the network, around 21 percentage of the nodes were labeled as “suspected as sybils”. The simulation result is shown in Fig. 5.

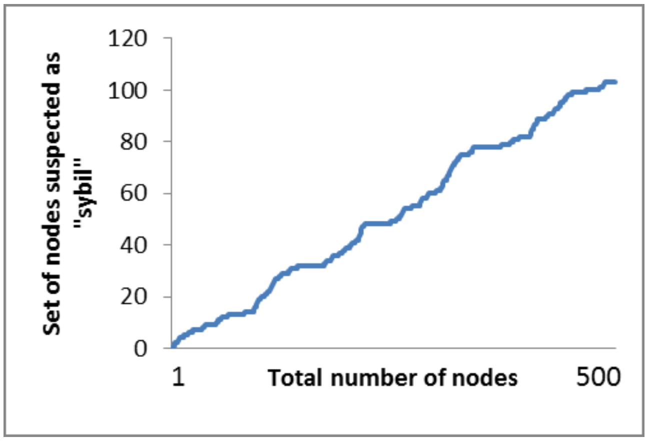

**Figure 5. Detection of Sybil Nodes**

False positives and false negatives are allowed in the labeling of a node as either “sybil” or “genuine”. So we can infer that the system will not consider those nodes which are suspected to be sybil for storing their replicated data in case of P2P backup systems or does not consider feedback from these nodes in case of P2P recommendation systems.

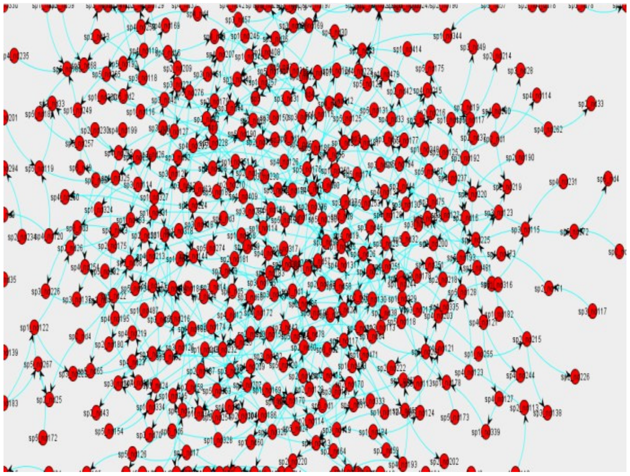

**Figure 6. Transaction between peers**

Meanwhile peers contact each other for file sharing. A node can contact with any other node in the system, even if it is a super node or nodes under any other super node. In our scheme, we put constraint only on node admission, so that once a node became part of the network, it can interact with any other node. Each peer maintains the history of all nodes with which a contact occurs if they come under the same super node. When peers contact, the one who got service, rate the other based on the authenticity of files, types of service etc. Figure 6 shows the transaction between peers.

When the number of nodes under any super node reaches 100 or its multiples, super node does the rank matrix calculation (Section 3.2). In our simulation setup, super node 1 (sp1) does the rank matrix calculation first. The number of nodes under sp1 is 100 and among them, by applying our algorithm we suspect 19 percentages of the nodes as sybils and rest as genuine. The values in final rank matrix calculation are considered for assignment of identities.

For each node, values in the rank matrix are considered row wise, and the these are added to get the Sum-direct rank matrix values and Sum- final rank matrix from direct rank matrix and final rank matrix respectively

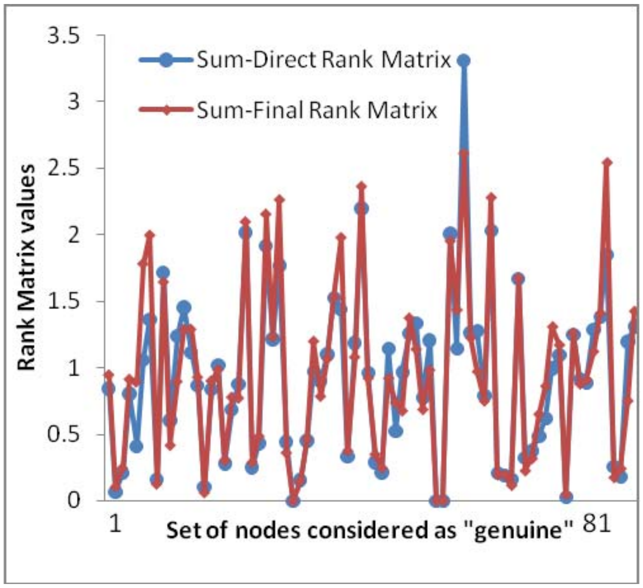

**Figure 7. Rankmatrix of nodes considered to be genuine**

The values in final rank matrix can be smaller, equal or higher than values in direct rank matrix. Usually it will be higher since values of indirect rank matrix are also considered while calculating final rank matrix. Here we put, $c = 0.4$ for conducting our simulation. From Fig. 7 we can infer that for most of the nodes considered to be genuine, final rank matrix values are equal to or greater than direct rank matrix.

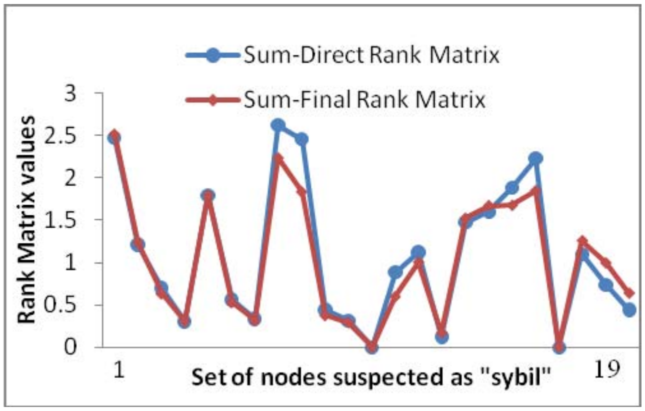

**Figure 8. Rank matrix of nodes suspected as sybils**

From Fig. 8 we can observe that, for most of the nodes which are suspected as sybils, value of final rank matrix is lower than value of direct rank matrix. Once super node computes the rank matrix, labeling of a node as either sybil or genuine is done by considering those values also. However, in labeling process, false negatives should be minimized as far as possible.

## 5. CONCLUSION

This paper proposes a simple mechanism for sybil resistant node admission in P2P networks. Using this scheme the sybil behavior of a node can be identified, and those suspected as sybils are limited from inviting others. Moreover nodes may contact one another for file sharing and super nodes calculate rank matrix and uses these values also for assignment of new identities. Although false positives and false negatives may occur in the labeling process, we have to minimize it as far possible, to improve the efficiency of the algorithm. In future, the efficiency of this algorithm can be increased by considering more parameters for labeling a node as sybil or genuine.

## 6. REFERENCES

[1] Wallach, D. S. 2002. A survey of peer-to-peer security issues, *In Proceedings of the International Symposium on Software Security (ISSS),* Springer-Verlag.

[2] RFC 4981 - *Survey of Research towards Robust Peer-to-Peer Networks.*

[3] Douceur, J. R. 2002. The Sybil attack*, In Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS),* Springer-Verlag.

[4] Yu, H. 2011. Sybil Defenses via Social Networks: A Tutorial and Survey, *SIGACT News*.

[5] Castro, Druschel, P., Ganesh, A., Rowstron, A., and Wallach, D. S. 2002. Secure routing for structured peer-to-peer overlay networks. *In Proceedings of 5th ACM Symposium on OSDI.*

[6] Levine, B. N., Shields, C. and Margolin. N. B. 2006, *A survey of solutions to the sybil attack.* Tech report, University of Massachusetts, Amherst.

[7] Borisov. N, 2006. Computational puzzles as sybil defenses*, In Proceedings of the 6th IEEE International Conference on Peer-to-Peer Computing,* IEEE Computer Society.

[8] Gatti, R., Lewis, S., Ozment, A., Rayna, T. and Serjantov, A. 2004. Sufficiently secure peer-to-peer networks*. In Workshop on the Economics of Information Security.*

[9] Yu, H., Kaminsky, M., Gibbons, P. B. and Flaxman, A. 2006. Sybilguard: defending against sybil attacks via social networks, *In Proceedings of the ACM SIGCOMM Conference.*

[10] Yu, H., Kaminsky, M., Gibbons, P. B. and Xiao, F. 2008. SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks*, In IEEE Symposium on Security and Privacy*.

[11] Yu, H., Kaminsky, M., Gibbons, P. B. and Xiao, F. 2009 DSybil: Optimal Sybil-Resistance for Recommendation Systems, *In IEEE Symposium on Security and Privacy.*

[12] Bazzi, R. and Konjevod,G. 2005.On the establishment of distinct identities in overlay networks, *In ACM PODC.*

[13] Lesueur, F., L. M´e and V. T. Tong. 2008. A Sybilproof Distributed Identity Management for P2P Networks, *In Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*, IEEE Computer Society, Morocco.

[14] Kesidis, G., Tangpong, A. and Griffin, C. 2009 A Sybil-proof Referral System, Based on Multiplicative Reputation Chains, *IEEE Communication Letters.*

[15] Kamvar, S. D., Schlosser, M. T. and Molina, H. G. 2003. The Eigen Trust Algorithm for Reputation Management in Peer to Peer Networks*, In Proceedings of the 12th international conference on World Wide Web*.