ISSN 2250-3153

Parental Control algorithm for Sybil detection in distributed P2P networks

Ankush Tehale, Amit Sadafule, Swapnil Shirsat, Rahul Jadhav, Satish Umbarje, Sandip Shingade

Information Technology, Veermata Jijabai Technological Institute, Mumbai, INDIA

Abstract- Distributed social peer to peer network are most vulnerable to Sybil attack. It forms a small network within the P2P network and can give unwanted results to other nodes in network, thereby decreasing the interest of non malicious nodes in the P2P network. We are proposing a *Parental Control* algorithm which is based on reputation scheme. It uses the *false message* concept for identifying and verifying the Sybil nodes in the network.

Index Terms- Reputation, peer to peer (P2P), social networks, DHT

I. INTRODUCTION

In Sybil attack[1] an attacker introduces itself in the network with many P2P identities. If an attacker gets large network identities, it can control large portion of the network. When an attacker wants to join the network, it most likely to get join its other fake nodes. Hence all Sybil nodes, most likely, form a small network inside the P2P network (in case of social network). But if identity assignment scheme, in P2P network, is *uniformly distributed*, then it is very difficult for an attacker to strategically place such Sybil nodes in network according to Castro [15].

Even such randomly placed attacking nodes can make harm to network. To avoid malicious activities in the network reputation scheme can be used. In reputation based schemes, all nodes get reputation based on the basis of their behavior in the network. If a node behaves well in the network its reputation will be improved. Using reputation based scheme some other attacks such as DDoS [16] can be prevented. But a Sybil node can increase its reputation by contacting to other Sybil node or it can defame the honest nodes. To minimize this effect, *referral system* [8] can be used. *DHTrust* [13] is a robust and distributed reputation system for Trusted Peer-to-Peer networks. It resists the malicious node from faking reputation.

This paper provides a model for Sybil detection in peer to peer network. This model has following components:

1. *Reputation scheme*: A robust reputation scheme is required for selection of suspects and trusted verifiers. Any reputation scheme can be used which is best suited to overlay network and most resistant to malicious activities.
2. *Verification Scheme*: Verifiers, selected using above scheme, are used for verification of nodes. Verifiers use this scheme for detecting the Sybil nodes.

II. TERMINOLOGY AND NOTATION

Table 1

Symbol	Meaning
N*i*	i-th node in the network
RM*i*	i-th Reputation Manager
LR*j*,*k*	Local reputation of node N*j* in reputation manager RM*k*
GR*i*	Global Reputation of node N*i*
T	Threshold on Global Reputation
QoS	Quality of Service
T*i*→*k*	Trust on RM*k* as observed by node N*i*
AQ	The average quantity of transactions of nodes in the system
RF	Reputation Feedback
P*i*	Parent of node N*i*
G	Group size a parent can handle

III. RELATED WORK

If Douceur [1] has proven that without use of central authority, it is not possible for a system to fully defend against Sybil attack. Hence in P2P network, which is fully distributed, Sybil nodes can not be removed completely from the network.

J. Dinger and H. Hartenstein [2] have given *Self Registration* mechanism. In this solution a node needs to register at 'r' successfully registered nodes in the network, and for verification any node checks its registration.

A. Cheng [7] has shown that no symmetric reputation function is Sybil proof. This paper has also given a general formula for asymmetric reputation function, which is more Sybil resistant.

G. Kesidis [8] have proposed a *Sybil-Proof Referral System*, which is based on Multiplicative Reputation Chains. Using multiplicative reputation chain, single step and multi step referrals can made Sybil proof. This technique is used in our solution to select more trusted node as verifier.

Haifeng Yu [4] provided *Sybil Guard* as solution to Sybil attack in social network. This solution restricts $O(\sqrt{n} \log n)$ Sybil nodes per attacking edge where n is number of nodes in the network. This solution uses pre computed permutations to generate random paths. Here, whenever an honest user's node is corrupted, the friends of that user could be targeted by the adversary to create a number of Sybil identities which Sybilguard will be unable to detect [5]. They further reduced the number of Sybil nodes per attacking edge to $O(\log n)$ in their next solution known as *Sybil limit*[3].

A.Kurve and G. Kesidis [9] provided *Sybil Detection via Distributed Sparse Cut Monitoring* as a solution. This method (also our) depends on the fact that, a Sybil cluster is typically sparsely connected via direct reputation links with the rest of the reputation graph [3][4]. It uses some trusted nodes as verifiers called as Police Nodes. Using min-sparse cut algorithm, it forms groups of nodes with nearly same reputation. The region with lower reputation is considered as Sybil group. For this a central trusted entity, which has entire network parameters, is required to form cuts in the network.

C. Hota [5] proposed *Safeguard* algorithm, some random verifiers are chosen. Each verifier verifies a group of random nodes, called as suspect group, by finding paths to each suspect node and then intersection of paths is taken. After intersection, the nodes remaining are more likely to be Sybils. To verify them further it used HIP [17]. But the main disadvantage is that HIP can not be used for verification if the node is under NAT (Network Address Translation).

IV. SYSTEM ARCHITECTURE

System is fully distributed peer to peer network. Nodes are connected to other nodes for means of communication. The architecture can be viewed as a graph G = (V, E), where V is set of vertices representing nodes and E is set of edges representing connection links between nodes (as shown in Fig.1).

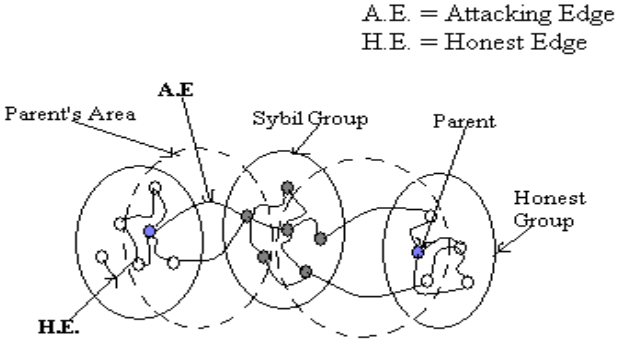

Fig.1: System Architecture.

Each node in V has unique identifier as per the P2P network. Each node has reputation in the network, which is decided dynamically based on its work in the network.

In social network, all Sybil nodes form a small network inside a network. Some of these Sybil nodes are connected to honest nodes. The edge connecting two honest nodes is called as *honest edge* and the edge connecting honest node to malicious node is called as *attacking edge*. Hence honest region is connected to Sybil region via attacking edges.

A distributed reputation scheme is deployed in the network. All nodes have a parent node P associated with them. Each parent has fixed set of nodes for under its consideration. There are more than one parent nodes in network.

V. DETECTION ALGORITHM

We are proposing an algorithm to detect Sybil nodes in the peer to peer network. To perform detection action we have provided additional functionalities to some selected nodes. These nodes in one particular group known as "parents" which is responsible to detect such suspect, attacked nodes in that particular group which is under his observation. This allocation of verifier to groups is done in such a way that there will not be more than one verifier in single group. Job of verifier is to find out group of suspect nodes which are sending false messages in the network i.e. Sybil nodes and keep the network Sybil proof and make network more reliable.

A. Reputation Scheme

Our detection algorithm requires a robust and Sybil proof reputation scheme present in the network. Distributed asymmetric reputation schemes [7] are resilient to Sybil attack. This scheme gives local reputations to each node, which can be used to calculate global reputation of the node. This global reputation used in the verification process. Sybil proof Referral System [8] can also be used to add more robustness. Only those nodes whose reputation is below threshold (T) are considered as suspect otherwise not.

Reputation of nodes is increased after each successful transmission of data and decreased after sending of each false message.

B. Detection Algorithm

Steps to detect suspect nodes are given as follows:

1. If any node say node N*i* requests for a file in the network and finds more than one nodes, who have the file, then N*i* selects one node out of all of these whose global reputation (GR) is higher. Let us say this node is N*j*.
2. If N*j* is Sybil attacker it will send a false message to node N*i*. Here by false message we mean wrong file sent; lower QoS than expected or no response from the sender. If N*j* is non malicious node, it may not send false message to node N*i*.
3. If N*i* receives false message, it complaints about the node N*j* to its parent P*i*.
4. Now parent will first find neighbors of node N*j* and put only those nodes in *suspect group* which has reputation below threshold (T) along with node N*j* (if node N*j*'s reputation is also below threshold).
5. To verify suspect group, P*i* will give a computational puzzle [14] to all these nodes. All the nodes have to give correct answer the puzzles in limited time simultaneously.
6. If there are Sybil nodes present in the suspect group it needs to respond to all puzzles from P*i* simultaneously but due to processing power limitation it will unable to respond in specific threshold time and is detected by P*i*.

Detection of Sybil nodes depends upon the size and other characteristics of puzzles [1]

VI. SIMULATION MODEL

We have used Chord [6], structured P2P network. DHTrust [13] is used as reputation scheme in the network. Though Chord is structured P2P network, SPROUT [12], a DHT algorithm, can be used to add social links in it. Entire network and algorithm is simulated on PeerSim [10] simulator. Network contains maximum 3000 nodes. Any node is selected at random after 20 simulation steps to send messages to other nodes. Total false messages (the unreliable messages sent by malicious node), total

malicious, total failed messages, network size are recorded every after 1000 simulation steps. Entire network is simulated for 1000000 simulation steps. Out of 3000 nodes in the network 10% nodes are malicious.

Before we move further let us discuss about working of DHTrust.

A. DHTrust

DHTrust selects several nodes as reputation managers to distribute the local reputation (LR) on them instead of putting the global reputation on score managers. This scheme is used on DHT based networks like Chord. In Chord network, according to DHTrust, each node in the Chord identifier circle selects all its finger nodes as its reputation managers. Local reputation of a node N*j* held by k-th reputation manager is LR*j,k*. Because the node of Chord maintains all its finger nodes in its finger table, it can retrieve its local reputation instantly according to its finger table. Assuming that the identifier circle has the size of $N = 2^M$, acquiring local reputation of a node to aggregating it into global reputation (GR) only needs O(logN) messages.

Suppose node N*i* wants to calculate the global reputation of node N*j*, it accesses the reputation manager nodes of node N*j* to obtain local reputation LR*j,k* ($1 \le k \le M$, M means the chord identifier's length). During the calculation of global reputation of node N*j* (GR*j*), trust on the k-th reputation manager with respect to the node Ni ($Ti \rightarrow k$) must be considered hence GR*j* is calculated as follows:

$$GRj = \frac{\sum_{k=1}^{M} (Sk \times Ti \rightarrow k \times LRj,k)}{\sum_{k=1}^{M} Sk} \quad (1)$$

Where S*k* is the size of the interval in which the *k*-th reputation manager is located. From eq. 1 it can be seen that global reputation is not dependent on a single reputation manager. Hence even if any reputation manager fakes about the reputation of node N*j*, it will have very small (even negligible) affect on GR*j*.

Any node can calculate global reputation before transaction and can select node with highest global reputation. After transaction the requesting node N*i* issues Reputation Feedback (RF) to update the local reputation of sending node N*j*. RF is the estimation of services. A normal node would issue it strictly in accordance with the quality of service (QoS), but a malicious node may generate a totally opposite one. N*i* can only modify the specified local reputation, which is the local reputation hold by the reputation manager of the interval which node N*i* belongs to. After locating specified reputation manager eq. 2 is used to update local reputation at this manager.

$$LR^n j,k = LR^{n-1} j,k + (1/AQ)\,(RF - LR^n j,k) \quad (2)$$

Where $LR^n j,k$ is the *k*-th local reputation of node N*j* being updated for n times, and $LR^{n-1} j,k$ is local reputation of last time. The AQ means the average quantity of transactions of nodes in the system. Here Ni also updates its trust vector by updating the trust value of the k-th reputation manager using following equation:

$$T^n i \rightarrow k = T^{n-1} i \rightarrow k + (1 - 2\,|QoS - LR^n j,k|) \times (1 - T^n i \rightarrow k) \times T^n i \rightarrow k / AQ \quad (3)$$

B. Sybil Detection in Chord

Different groups are formed in the Chord network with the fixed group size (G). Each group has a parent node. Here we have chosen random nodes as parents.

Suppose node N*i* has done transaction with another node N*j* and Ni received a *false message* (as described above) it signifies this to its parent (P*i*) by issuing a *Detect Message.* After receiving this message P*i* adds nodes N*j* and all finger nodes of N*j* in suspect list of Pi only if GR of the nodes is less than *Threshold* (T) reputation. Now P*i* applies verification algorithm to this suspect list.

As we have used puzzle based verification, hence the efficiency of the Sybil Detection Algorithm depends on puzzle solving time and Threshold (T) on global reputation. If T is very small, Sybil detection will be very hard. If T is very high, Sybil nodes need to maintain reputation higher than T, to be undetected, but for this Sybil nodes must do trusted and correct transactions. Higher value of T also increases the overload on parent node P*i* as this increases the size of suspect list. Hence T much be chosen so that to obtain overall efficiency.

Sybil node detection also depends on suspect group size (SG). This dependency is described by following graphs which are drawn assuming malicious entity has 4GHz processor and the puzzle is chosen so that it requires 1sec to solve to a 1GHz processor. Also it is assumed that in this social network each node has maximum 20 social links.

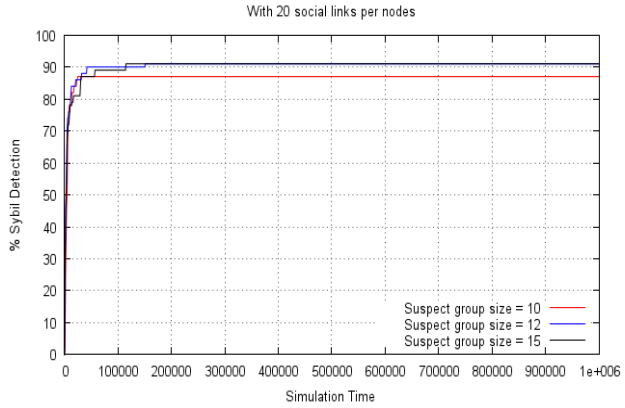

Fig.3: % of Sybil nodes detected as simulation time passes

From Fig.3, we can see that efficiency of this algorithm depends on suspect group size. For SG=15 and SG=12 Sybil nodes, this algorithm has about 90% efficiency and for SG=10 Sybil nodes, it has about 87% efficiency. Hence it shows as SG increases efficiency of Sybil detection increases. From Fig.4, it can be seen that with increase in suspect group size, total number of false messages in the network are also reduced. For SG=10, SG=12 and SG=15 Sybil nodes, total false messages are reduced by about 85% to 90%.

Number of social links per node in the network also has effect on this Sybil node detection algorithm. It can be shown by Fig.5. It can be seen that efficiency of the algorithm increases with increase in number of social links per node. This is because; it allows more neighbor nodes of a node to be considered as Sybil nodes when the node sends false messages.

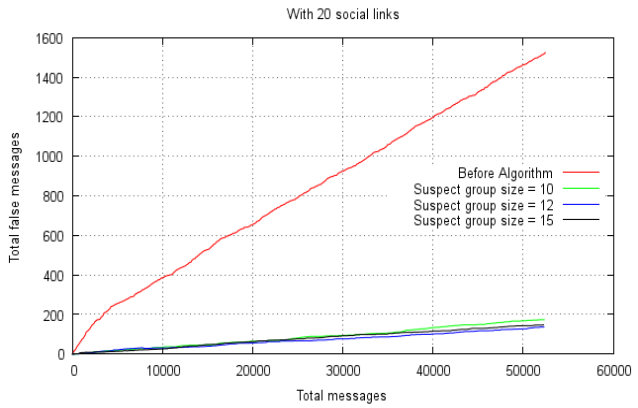

Fig.4: Total Transaction messages Vs total false messages before and after Sybil detection.

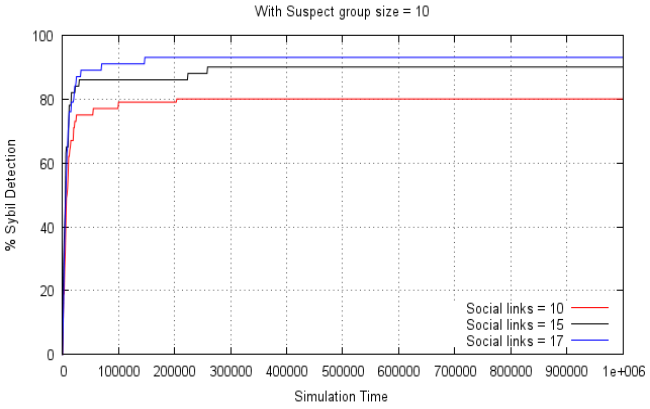

Fig.5: Change in % Sybil detection with social links.

VII. CONCLUSION

We presented a parent based detection algorithm for Sybil nodes. This algorithm detects about 90% of Sybil nodes in the static network as well as false messages are decreased by 85% to 90%. As Sybil nodes form a small network within the peer to peer network, hence, whenever a Sybil node issues false message to non Sybil node, the Sybil node along with its neighbors are verified. Due to this property of algorithm more Sybil nodes are detected at a time.

Future work includes, implementing the algorithm on dynamic network, selecting repudiated and trusted node as parent to newly arrived nodes. Also, parent node may go off at any time hence new trusted parent must be allocated to the group that it was guiding.

REFERENCES

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

[2] 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.

[3] 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.*

[4] 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.

[5] 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

[6] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications", In *Proc. 2001 ACM SIGCOMM Conference*, ACM Press, 2001, pp. 149–160.

[7] 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.

[8] 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.

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

[10] Jelasity, M., Jesi, G.P., Montresor, A., Voulgaris, S.: PeerSim P2P Simulator. http://peersim.sourceforge.net/

[11] X. Yue, Xiaofeng Qiu, Yang Ji, C.Zhang, "P2P Attack Taxonomy and relationship Analysis", in *Proc. Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference*, pp. 1207 – 1210.

[12] S. Marti, P. Ganesan and H. Garcia-Molina "SPROUT: P2P Routing with Social Networks" *Current Trends in Database Technology - EDBT 2004 Workshops*, pp. 511-512.

[13] Y. Liu, W. Xue, K. Li, Z. Chi, G. Min, W. Qu, "DHTrust: A Robust and Distributed Reputation System for Trusted Peer-to-Peer Networks", in *Proc. Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1-6.

[14] Nikita Borisov, "Computational Puzzles as Sybil Defenses", in Proc. Peer-to-Peer Computing, 2006. P2P 2006. Sixth IEEE International Conference, pp. 171 – 176.

[15] M. Castro, P. Druschel, A. Ganesh, et al., "Secured Routing for Structured Peer-to-Peer Overlay Networks", In *Proceedings of 5th Usenix Symposium on Operating Systems Design and Implementation*, Boston, MA, December 2002, pp. 299-314.

[16] Ankur Gupta , Deepti Malhotra, Lalit K. Awasthi, "NeighborTrust: A Trust-based Scheme for Countering Distributed Denial-of-Service Attacks in P2P Networks", in *Proc. Networks, 2008. ICON 2008. 16th IEEE International Conference*, pp. 1-6.

[17] R. Moskowitz, and P. Nikander, "Host Identity Protocol (HIP) Architecture", RFC 4423, May 2006.

AUTHORS

First Author: Ankush Shambhudev Tehale, pursuing B.Tech. Information Technology in Veermata Jijabai Technological Institute, Mumbai, INDIA.
Email id - ankushtehale.vjti@gmail.com.

Second Author: Amit Sanjay Sadafule, pursuing B.Tech. Information Technology in Veermata Jijabai Technological Institute, Mumbai, INDIA,
Email id - amit.sadafule@gmail.com.

Third Author: Swapnil Kashinath Shirsat, pursuing B.Tech. Information Technology in Veermata Jijabai Technological Institute, Mumbai, INDIA,
Email id - swapnil.k.shirsat@gmail.com.

Fourth Author: Rahul Mohan Jadhav, pursuing B.Tech. Information Technology in Veermata Jijabai Technological Institute, Mumbai, INDIA,
Email id - rahulash.987@gmail.com.

Fifth Author: Satish Mahadev Umbarje, pursuing B.Tech. Information Technology in Veermata Jijabai Technological Institute, Mumbai, INDIA,
Email id - satishmaddyy @gmail.com.

Sixth Author: Sandip Shingade, Assistant Professor at VJTI Collage, M.Tech., Computer Science and Engineering from VJTI, BE, Computer Engineering from Vishwakarma Institute of Technology, Pune,
Email id - stshingade@vjti.org.in.