Abstract—The recent surge in popularity of peer-to-peer(P2P) applications like e-commerce have driven new interest in usage of reputation systems as a mechanism of managing trust between its users. Since these applications span the Internet, the probability of presence of adversaries who try to compromise the system is very high. The open and anonymous nature of P2P systems paves the way for malicious users to abuse the systems. A very convenient way for such malicious users to subvert the system is to compromise the identity generation process which forms the core of any successful reputation system. A single malicious user can generate multiple identities to falsely raise its reputation value by providing good recommendations to itself through its sybils thereby launching "sybil attack". The robustness of applications built over P2P systems and in particular structured overlays suffers significantly when they are subjected to sybil attack.

In this paper, we propose a new strategy wherein the presence of sybils can be detected in reputation systems build over structured overlays. In our approach, each user plays a pivotal role in monitoring the message traffic behavior of its neighbors. The message traffic analysis of a peer gives rise to many interesting message patterns which leads to successful detection of the presence of sybils on its neighbors. Once the sybil attack is detected, we need to reduce its impact on the overall robustness of P2P reputation systems. In order to do so, we present a novel approach to isolate users who launch sybil attack thereby mitigating the threat imposed by them.

I. INTRODUCTION

The characteristic features of peer-to-peer (P2P) networks that are responsible for their tremendous success are their open, anonymous and decentralized nature. However these features pose a serious challenge to the security of applications which run over very large scale P2P systems like Internet. Most of the transactions in such applications happen between users who are virtual strangers to each other. It is very difficult to figure out the 'trustworthiness' of other users and the chances of getting cheated is very high. In such a scenario, reputation systems helps in making an informed decision regarding the 'trustworthiness' of other users based on their reputation[1]. The success achieved by online auction sites like eBay[2] and Amazon[3] which uses reputation systems to prevent frauds is a glowing example to their usefulness.

In a reputation system, each user is associated with a reputation value which denotes the opinion of other users about its trustworthiness. It is a reflection of user’s behavior in the past and also an indicator to its reliability in future transactions. Good users with high reputation values are rewarded. Malicious users with low reputation values are made to pay the penalty associated with their bad behavior. Each reputation system has its own way of implementing rewards and penalties for its users. In order to escape this penalty, malicious users try to subvert the reputation system by launching various types of security attacks like badmouthing[4], shilling attack[5], ballot-stuffing[4], etc.

In a reputation system, each user’s reputation value is associated with its identity. Since entry barrier is very low in P2P systems[6] and presence of zero-cost identities[7] help users in creating new identities easily. As a result a single malicious user can easily subvert reputation system by creating large number of identities called ‘sybils’ and by falsely raising its reputation value through fake transactions between these sybils. This is known as "Sybil Attack"[8].

In order to build a sybilproof reputation system one of the following two approaches has to be adopted:

1) Generation of sybils should be restricted through identity assignment processes.
2) Presence of sybils should be detected through identity verification strategies and then their impact should be minimized.

Decentralized nature of P2P systems necessiates the need for self generation of identities by peer themselves[6], [9], etc). This may lead to sybil attack since a single peer can generate as many identities as it wants. Considerable amount of research has been done regarding identity generation strategies[6], [8], [9]). However Douceur has shown the fundamental restrictive nature of such distributed sybil-proof systems[8]. Most of the research work with respect to identity verification strategies[8], [10], [11]) employ various resource consuming proofs in order to distinguish between identities and hence detect the presence of sybils. However in [8], Douceur has acknowledged the inherent limitations of identity verification strategies which employ resource consuming puzzles to detect the presence of sybils. In addition to this, huge disparities in the amount of resources available to peers
in heterogenous P2P systems render such methods ineffective. Moreover these approaches place heavy burden on peers since substantial amount of time and efforts are required for successfully cracking these resource based challenges. In [9], the defence against sybil attack is based on the fundamental premise that a peer cannot have access to large number of discontinuous IP spaces. However the main drawback of this approach is the identity generation process which assumes the presence of some non anonymous pre-existing trusted peers.

In this paper, the focus is on detecting the presence of sybils through verification of user identities in P2P reputation systems. In order to overcome the limitations of existing identity verification strategies, we present a novel approach of detecting the presence of sybils by analysing the message traffic data of each peer. To the best knowledge of the authors, the study of message traffic data to recognise message patterns which helps in defending against sybil attack has been not been proposed. In our method, each peer plays an important role in monitoring the message traffic of its neighbors which helps in detecting the presence of sybils on them. Our approach is very generalised in nature and doesn’t make any assumptions regarding the type of underlying structured overlay (like [12], [15]). Moreover this sybil detection strategy can be integrated with any reputation system built over structured overlays like [15].

The rest of the paper is organized as follows: Section II gives a brief overview of our system model and related assumptions. Section III explains the threat model that we assume while proposing the sybil detection strategy. In Section IV, we present details regarding the neighbor monitoring activities of peers which leads to detection of sybil attack. In this section we also present a way of isolating adversaries who launch sybil attack thereby minimizing the impact of sybil attack on P2P reputation systems. We present our conclusion in Section V along with future research directions.

![Fig. 1. P2P Overlay network architecture](image)

**II. SYSTEM MODEL**

Researchers in general use the term peer to refer to both human users and nodes interchangeably. However there is a clear distinction between these two entities. Nodes are present at overlay layer and users (residing on nodes) operate at application layer as shown in Figure 1 and also mentioned in [6]. In this paper the term peer refers to human users since we are aiming at detecting the presence of sybils at user level for an application (like ecommerce, file or resource sharing applications) which incorporates reputation system built over structured overlay. Hence the terms peer and user are used interchangeably throughout this paper.

In P2P systems and hence in structured overlays, the nodes are identified based on their node identifiers. Extensive research work has been carried out regarding the node identifier assignment process (as in [8], [6], [16]). In [6], it has been suggested that any reasonable P2P reputation system depend on the underlying P2P networks that are robust against sybil attacks. So in our model, it is assumed that the underlying structured overlay is pre-existing and the node identifiers are generated in a sybil-free manner such that each node has only one node identifier associated with it.

In a reputation system, the users are identified based on user identities which can be generated either centrally or in distributed manner. However in P2P reputation systems, the preferred approach is self generation (as in [6], [9]). In our model, the users of reputation system are identified by a public-private key pair and each UserId is self generated by the user. This prevents peers from masquerading as other peers with good reputation values. In addition to this, public-private key pair helps in maintaining the anonymity of users of P2P system.

In P2P systems, messages get routed based on the target node’s NodeId. In our strategy, we place an additional constraint that any message exchanged between users of reputation system contains both NodeId as well as UserId. It should be noted that since messages contain NodeId, there will not be any changes to the existing routing mechanisms of underlying structured overlays. The presence of UserId helps in tracking the message traffic for any user.

In our approach, each peer plays a role in monitoring the message traffic of its neighbors. However it should be noted that in our approach, the term neighbor nodes refers to immediate one hop neighbor nodes at the logical overlay layer and not to the entire set of neighbor nodes associated with overlay routing. The details regarding this message traffic analysis of peers which leads to detection of sybil attack is explained in detail in Section IV.

**III. THREAT MODEL**

In our system model, each user identity is self generated. As a result, malicious users of the reputation system may generate as many user identities as they want thereby leading to sybil attack. They do so in order to raise their reputation values by creating false transactions between these sybils or to launch other types of security attacks like eclipse attack[17], badmouthing attack[4], shilling attack[5], etc.

Self generation of user identities can lead to multiple UserIds associated with a node in the overlay. Since public-private key pair is used as UserId in order to maintain anonymity of users, it becomes very difficult to verify if different user identities refer to different legitimate users of the system or sybils created by a single malicious user. The situation becomes worse when a group of malicious users collude together thereby launching collusion attack[7].
In P2P systems, each user resides on a node in the overlay. In general, a user can own/use single or multiple nodes in the overlay network. Similarly a single node can be owned/used by a single user or shared by a group of users. The resulting identity space is shown in Figure 2 along with the mapping relation that exists between each of these identity spaces. In our current model, we assume that a single user does not own/use more than one node in the overlay. In other words, we are restricting m:n mapping between user space and node identity space to n:1 which in turn reduces the mapping relation between node identity space and user identity space to n:1 as shown in the Figure 2.

![Fig. 2. Relation between User, User Identity and Node Identity spaces](image)

It is possible that a single user can own a group of nodes in the overlay. However it should be noted that a single malicious user owning multiple nodes is equivalent to a group of distinct malicious users each residing on separate nodes and colluding together to subvert the reputation system. The combination of collusion and sybil attack is a complicated problem which is acknowledged in [7]. Hence we assume that each adversary use/own one node and acts alone. The scenario arising out of a user using a group of nodes and also the effects of collusion of different malicious users on our strategy is considered as part of our future work.

IV. PROPOSED SYBIL DETECTION STRATEGY

A. How a peer monitors its neighbors?

P2P networks are logical overlays built on top of existing physical networks like Internet. Structured P2P overlays (like [12], [13], [14]) impose constraints on the topology of the network. Hence for any node in the structured overlay, the set of neighbor nodes is well defined. The criteria for neighbor selection depends on the type of underlying structured overlay. Hence it is assumed that each peer will know the set of its immediate one hop neighbor nodes. As mentioned in Section III, each peer uses/resides one node and since collusion attack is assumed to be nonexistent, the users on immediate neighbor nodes can be expected to impartially trace the message traffic of that user.

Any message sent to a user or originating from that user residing on any node in the network has to get routed through one of its ‘k’ immediate neighbor nodes with probability 1/k. Hence the users on these immediate neighbor nodes play a crucial role in monitoring the message traffic of any user in the system. As part of this monitoring process, each user maintains a table called NeighborStatsTable. Since each message carries both NodeId as well as UserId (refer Section II), these details along with message arrival time are logged in NeighborStatsTable for every message originating from or meant for immediate neighbors. The exact mechanism in which the contents of NeighborStatsTable of immediate neighbors of a peer helps in determining presence of sybils has been explained in Sections IV-B and IV-C.

Since P2P systems are dynamic in nature and NeighborStatsTable data help a user in detecting presence of sybils, its contents should remain fresh for each peer. In order to do so, each entry remains in this table for only some time duration t after which it would be erased. The threshold value for the time duration t can be set by each user. We can observe that any changes in the neighbor set of a peer invalidates some of the contents of NeighborStatsTable and hence it would take longer time for a peer to detect the presence of sybils on its immediate neighbors. So the threshold value for time duration t is directly dependent on the changes in immediate neighbor set of a peer which in turn depends on churn rate of P2P networks. Thus each peer can set this threshold value depending on the churn rate of the system.

The size of NeighborStatsTable of any peer depends on the number of its immediate neighbors nodes, the number of UserId associated with each of these neighbors, the frequency of message arrival/departure of each of these neighbors and the time duration t set by each peer. However the size of overall neighbor set of peers in DHT based structured overlays is approximately around O(log N) where N is size of the network[18] and the immediate neighbor set of a user is a subset of this neighbor set. Hence this limits the growth of the size of NeighborStatsTable of any peer.

The proposed approach of message traffic monitoring process is explained in detail with the help of an example. Consider the snapshot of a part of sample structured overlay as shown in Figure 3. We can observe that the immediate neighbors of the node 112431 includes {112430, 112433, 112434, 112435, 112436, 112437} and the immediate neighbor set of node 112433 includes {112430, 112431, 112437}. The identities of users residing on corresponding nodes are shown in the figure.

![Fig. 3. Snapshot of a part of structured overlay where user on 112431 is malicious](image)
A and G. This may refer to a case of two legitimate users with identities A and G sharing the same system or a case of a single malicious user generating two sybils A and G. As part of the monitoring process, the immediate neighbors of 112431 trace its message traffic. Let’s consider one such neighbor 112433. The contents of NeighborStatsTable of peer on 112433 would be as shown in Table I at some instance of time. We can observe that the contents of NeighborStatsTable of any neighbor alone is not sufficient to detect the presence of sybils. Further probing is necessary in order to predict the presence of sybils which is explained in detail in Section IV-B and IV-C.

### B. How to identify sybil attack and reduce its impact?

In [19], a generalized lookup procedure is explained for P2P file sharing applications which uses reputation of peers before retrieving required files from them. In this section we extend the same to present a very generalised method for performing any type of search based transactions between application users. Here the term transaction is used to refer to any type of interaction between users of P2P reputation systems. It could be sharing of files in file sharing applications, sharing resource in resource sharing applications or financial transactions in ecommerce applications. In addition to this, our sybil detection procedure is integrated to isolate malicious users who launch sybil attack as shown below:

1. A requester peer queries for the provider peers (like resource or file owners, potential vendors, etc).
2. All the replies from provider peers are collected to form ProviderPeerSet.
3. A sybil-free provider peer with good reputation value is chosen from ProviderPeerSet as follows:
   a) A provider peer with highest reputation value is chosen from ProviderPeerSet.
   b) DetectSybilProcedure is run to check if this provider peer has launched sybil attack.
   c) If yes then lower the reputation of this peer. Then discard this provider peer from ProviderPeerSet and go to step 3a. If no then go to step 3d.
   d) Perform the transaction with this provider peer.
   e) If the transaction is successful, increase the provider peer’s reputation value and terminate the transaction process successfully. If the transaction is a failure, then decrease the provider peer’s reputation value. Discard this peer from ProviderPeerSet and goto step 3a.

Step 3 is repeated until either the desired transaction is successful or no more sybil-free peer with good reputation is available and hence the the transaction is never carried out thereby protecting the requester peer from frauds.

During DetectSybilProcedure, if it is known that a provider peer is malicious in nature, it is ignored by requester peer for all future transactions. Moreover by maintaining a local repository of such malicious user details, subsequent checking would be simplified. In this way, a reputation system is protected from being subverted by these malicious users. It should be noted that no assumptions are made regarding the type of reputation system and the mechanism of raising or lowering reputation values of peers. Hence our method can be integrated with any type of reputation systems build over structured overlays like [15].

### C. How to implement DetectSybilProcedure?

The DetectSybilProcedure mentioned in Section IV-B is the key procedure which informs a requester peer about the probability of a provider peer being involved in sybil attack. It can be implemented as shown below:

1) Generate a request to get the contents of NeighborStatsTable of each neighbor corresponding to provider peer’s node identity.
2) The neighbors of provider peer send their NeighborStatsTable data to requester peer.
3) Analyse the patterns arising out of the message traffic data of each neighbor’s NeighborStatsTable contents and predict the probability of provider peer being involved in sybil attack.
4) If the probability of the occurrence of sybil attack predicted in previous step is greater than the threshold value set by the requester peer, return yes. Else return no.

The step 1 of DetectSybilProcedure can be implemented in one of the following ways:

**Method 1:** The requester peer sends a request message to the selected provider peer requesting for its neighbors’ information which includes both NodeId as well as UserId of all its neighbors. Once the provider peer responds with a set of ‘neighbor’ information, requester peer has to verify this list since a malicious provider peer may return information about its sybils and not actual neighbors’ information. Some mechanisms like resource based puzzles should be employed in order to detect if these neighbors are legitimate peers or sybils as mentioned in [8] and [11]. Once the neighbors are verified, each of them are individually sent a message requesting for their NeighborStatsTable contents corresponding to provider peer node identity.

**Method 2:** The requester peer sends a request message to the selected provider peer requesting for its neighbors’ information. This message would get routed to one of its neighbors before reaching the provider peer. Instead of just forwarding this message to selected provider peer, the neighbor peer (which receives this message) would flood this request message locally so that this request message reaches all the neighbors of the provider peer.

<table>
<thead>
<tr>
<th>NodeId</th>
<th>UserId</th>
<th>Message Arrival Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>112430</td>
<td>E</td>
<td>T3</td>
</tr>
<tr>
<td>112431</td>
<td>A</td>
<td>T1</td>
</tr>
<tr>
<td>112432</td>
<td>G</td>
<td>T4</td>
</tr>
<tr>
<td>112437</td>
<td>C</td>
<td>T2</td>
</tr>
</tbody>
</table>
We can observe that Method 1 requires \((1+k)\) messages from requester peer where \(k\) is the number of immediate neighbors of provider peer who are involved in monitoring process. But Method 2 requires a single message from requester peer. Moreover in Method 1, requester peer has to verify if the ‘neighbor’ information returned by provider peer is valid or not but in Method 2, no such verification is required since the neighbors of provider peer are contacted directly without the help of provider peer. However Method 2 involves local flooding by one of the neighbors such that the request message reaches all the neighbors of provider peer.

The step 3 of DetectSybilProcedure is the most crucial step of this procedure which predicts the probability of presence of sybils. Various message patterns could arise out of this analysis of NeighborStatsTable of each neighbor. Currently we have presented only two very basic message traffic patterns in this paper. Work is in progress regarding the detection and analysis of other message patterns. These message patterns are explained below:

1) **Pattern 1**: In each of the NeighborStatsTable of immediate neighbors, only one user identity is logged against the node identity of provider peer at various message arrival times.

2) **Pattern 2**: In each of the NeighborStatsTable of immediate neighbors, more than one user identity is logged against the node identity of provider peer at various message arrival times.

If Pattern 1 is observed then this refers to a scenario wherein the provider peer (with only one user identity) was using the node referring to provider peer node identity. In other words, provider peer is the sole occupant of the node on which it resides. Hence this can be inferred as legitimate and hence non-sybil case.

If Pattern 2 is observed then this refers to a scenario wherein multiple user identities were using the node corresponding to provider peer’s node identity. In other words, the node on which provider peer resides could have: (a) multiple legitimate users sharing the system or (b) a single user (ie provider peer) who has created multiple sybils or (c) combination of both. If it is case (c), then malicious users with sybils and legitimate users co-exist simultaneously and provider peer could belong to either group.

In such a scenario, the past transaction history in the reputation system associated with each of these user identities are verified to check if all these UserIds are involved in providing good recommendations to other UserIds associated with NodeId of provider peer. Recall that in our threat model (Section III), we made an assumption that each user would use/work on one node in the overlay. If so then this strongly refers to sybil attack and hence more weightage is given for this (under the assumption that sybils would be interested in raising their own reputations rather than bad-mouth others as assumed in [20]). However if these user identities are involved in transactions with other user identities located on other nodes then this strongly hints at the case of a group of legitimate users sharing the system and hence more weightage is given to this being legitimate non-sybil case.

In case (c), the presence of sybils of one malicious user and other legitimate users further complicate the analysis process. However in this case also, the past transaction history of these user identities play a key role during inference step. From this past history, we can predict which group of user identities are involved in providing good feedback to each other within the group and other user identities which are involved in normal transactions (which includes some failed transactions and some transactions with users on other node’s user identities). However this case is much harder to infer compared to cases (a) and (b).

We can observe that during inferencing on Pattern 2, we rely on the availability of past history of transactions in the reputation system. However when no prior transaction history is available, then it would be difficult to differentiate between sybil attack and legitimate case of multiple users on same node. In such cases, our inferencing procedure relies on message pattern analysis alone.

When message traffic data from each neighbors of a peer is analysed, many interested patterns were detected two of which we have presented in this section. We are in the process of trying to detect more patterns which help us in our inference process. Further we need to simulate our algorithm and check to what extent our approach of message pattern analysis can withstand sybil attack.

### V. Conclusions and Future Work

In this paper, we have presented a new strategy for detecting the presence of sybils based on the message traffic analysis of peers in reputation systems built on top of structured overlay. We have identified two very basic message patterns that arise out of this message traffic analysis by neighbors of a peer. We have also presented a way for integrating our sybil detection strategy with any existing P2P reputation system. Further we have shown that it is the immediate neighbors of a peer who can effectively trace the sybil attack.

There are many interesting research directions which have opened up with respect to this line of work. We can further study message traffic data and detect more message patterns which strengthen our inference process in diagnosing sybil attack (even in the case of a user using multiple nodes in the P2P network). In our sybil detection strategy the set of immediate neighbors who perform message traffic analysis of a peer is chosen based on the structured overlay topology. However if the overlay topology and hence the neighbor set of any peer changes continuously i.e. if there is high churn rate in the P2P system, then it partially invalidates the contents of NeighborStatsTable of peers and hence the time taken for any peer to detect the presence of sybils increases. In addition to this we can experimentally study the effects of churn on the success rate of the proposed sybil detection strategy. Furthermore we can also study the effects of time on the freshness of NeighborStatsTable data of any peer.
REFERENCES


