Containing Sybil Attacks on Trust Management Schemes for Peer-to-Peer Networks

Lin Cai and Roberto Rojas-Cessa

***Abstract*—In this paper, we introduce a framework to detect possible sybil attacks against a trust management scheme of peer-to-peer (P2P) networks used for limiting the proliferation of malware. Sybil attacks may underscore the effectivity of such schemes as malicious peers may use bogus identities to artificially manipulate the reputation, and therefore, the levels of trust of several legitimate and honest peers. The framework includes a k-means clustering scheme, a method to verify the transactions reported by peers, and identification of possible collaborations between peers. We prove that as the amount of public information on peers increases, the effectivity of sybil attacks may decrease. We study the performance of each of these mechanisms, in terms of the number of infected peers in a P2P network, using computer simulation. We show the effect of each mechanism and their combinations. We show that the combination of these schemes is effective and efficient.**

***Index Terms*—Distributed system, p2p network, sybil attack, malware proliferation, key mean clustering, transaction verification.**

I. Introduction

Trust management is a proactive mechanism used to associate the quality of services that cooperative peers provide to other peers in a network [1], [2]. These mechanisms have been adopted as an effective measure to prevent the dissemination of malware in peer-to-peer (P2P) networks [3]–[11]. Existing trust models evaluate a peer's reputation by considering the transactions experienced and reported by interacting peers. A numerical reputation is not only associated to peers, but also to the objects exchanged, such as files. Peers that experience object acquisition evaluate the final outcome and publish it as public feedback. In these approaches, a peer's reputation is part of the score that each peer estimates about the peers who provide files [4], [8], [9]. Each peer builds a reputation score for each of its interacting peers (and other not interacting peers), and it is used to estimate a trust value. The reputation scores and trust values are used by a peer to decide whether or not to interact with the evaluated peer, and to decide whether the acquisition of the object may be pursued.

A reputation system can be roughly classified as synchronous or asynchronous. In a synchronous reputation system, such as EigenTrust [5], the trust value is calculated from the values estimated by all the peers in the network. The identification of trustable peers is achieved through the dissemination of their reputation values by participating peers. The topology of the network formed by the group of interacting peers also affects the scores, which are consolidated in a trust table. In these systems, reputation for each peer is calculated locally (i.e., each peer calculates its own trust table) and independently from other peers' tables.

However, an attack on the reputation system may come in the form of a sybil. In a sybil attack, malicious peers may take several identities and report fictitious interactions, with the objective of artificially building up a sybil's reputation [12]–[18]. The goal of such an attack is to manipulate the reputation of nodes or to influence the trust on particular peers. For example, a sybil attack could be used to steer searches or network traffic towards a malicious peer for exploitations.

Under such an attack, a trust management system may lose effectivity. A single peer may be able to affect the trust estimations of a network [19]. As our concern is the proliferation of malware through file exchanges, the higher the reputation of a peer, the larger the number of downloads may be sought from it. We set as the final goal of a sybil the release of malware effectively.

Sybil attacks are difficult to detect. Take for example a case where honest peers start interacting with the sybil, it then becomes complex to identify the sybil. Existing defense schemes against sybil attacks consider different tradeoffs and show that sybil attacks are difficult to thwart [3], [5], [19].

Synchronous reputation systems may be highly vulnerable to sybil attacks [19]. In such systems, a sybil may impersonate peers by creating a copy of a (trust) relationship graph. This graph may be used to report transactions favorable to sybils, and therefore, it may be used to quickly increase their reputation, in a way that otherwise would be seem honest. Therefore, honest peers may not be differentiated from sybils in such a graph.

On the other hand, asynchronous reputation, where each peer builds its trust values locally and independently, systems may be more robust to sybil attacks because no sybil may base its estimations to a global (and centralized) graph. However, such a network may still be subverted but by a large number of sybils. As discussed before, both reputation systems may be affected by reporting fictitious transactions. This issue raises the following question: would the detection of sybils be more effective than the use of methods to verify reported transactions for avoiding the subversion of a trust management scheme by a sybil attack?

Therefore, we introduce a framework for containing the proliferation of malware in P2P networks under sybil attacks. For framework design, we identify a set of necessary conditions to make a trust management scheme effective against sybil attacks. A general parameter is the amount of peer information for identification of sybils. We show that as the amount of information increases, being quantifiable, so does the effectiveness of a defense mechanism agains a sybil attack.

The authors are with the Networking Research Laboratory, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA. Email: rojas@njit.edu.

978-1-4799-2003-7/14/$31.00 ©2014 IEEE

Furthermore, we propose three different defense mechanisms: 1) A local trust table, where peers record the different trust values, one per peer, and their collaborations with other peers. 2) A k-means clustering mechanism to differentiate honest peers from sybil peers, without recurring to threshold based schemes [5], [7], as the selection of suitable threshold values is complex. 3) A transaction verification scheme to verify the occurrence of reported interactions between peers.

The remainder of this paper is organized as follows. Section II describes the model of a sybil attack and presents a preliminary analysis on the requirements for increasing the effectivity of a trust management scheme. Section III introduces the peer clustering and local trust table mechanisms. Section IV introduces the proposed transaction verification mechanism. Section IV-A introduces the trust management model. Section V shows the performance results obtained through computer simulation. Section VI presents our conclusions.

II. Trust Management Scheme and Problem Statement

Sybils attempt to increase the trust value of a single or multiple sybils to make them attractive (highly trustable) sources of files. Once a sybil is accepted for interaction with an honest peer, it may release malware and infect the honest peer.

A. Peer-to-Peer Network and Sybil Attack Models

The vulnerability of a trust management system under a sybil attack is determined by how effectively sybils affect the trust values on sybil and honest peers. In this network, a peer trusted by peer i is called *trustee*, which is the source of a file, of peer i, and peer i is called *truster* of that peer. Peer i has a trust table, which is denoted as $T(i)$. The trust value of peer i about peer j, $T_v(i,j)$, indicates the number of malware-free (or clean) downloads divided by total number of downloads. Each download may be the transfer of a complete file. After a download, the truster peer reports the experienced quality of the download to its trustees. The message includes the truster identification and it is called a rating message. Peers receiving this information use it to adjust their trust values about the reported peer(s).

A rating message is reported as either positive or negative. The message is propagated through a chain of trustees, and the social distance (which is equivalent to the number of peers the message traverses from the original source) affects the weight of the adjustment. After peer j receives the message, it calculates the propagated value $P_v(j,k)$, which is the average of the propagated ratings about peer k. To avoid having any peer being able to affect indiscriminately large number of peers, peer j accepts rating messages only from its trusters. Each peer calculates a trust value and a propagated value on each peer from its downloading history and the rating messages. Therefore, each peer calculates trust values locally and independently.

B. General Model of Sybil Attack on a Trust Management Scheme

In this section, we explore the necessary conditions for making a trust management scheme effective despite being under sybil attacks. We consider a discrete time system, where each event, such as a download, the sending of a rating message, or the search for a file, occurs in a time slot.

Consider a network with n honest peers and s sybil peers, each honest peer has an average number of trusters, m, and each sybil peer has an average of r sybil identities (i.e., sybil peers). The total number of peers in the network is $N = n + s + sr$.

Let $I(t)$ represent the number of infected peers (i.e., honest peers that have downloaded malware, and who may inadvertently disseminate the malware to other peers) in the network at time slot t, and let $T(t)$ represent the number of honest peers in the P2P network at time slot t. Therefore,

$$T(t) + I(t) = n + s + sr \tag{1}$$

Each peer performs a download at time slot t with probability p. The total number of downloads in a time slot is $(n+s+sr)p$, and the probability that a download is performed from a sybil peer at time slot t is γ_t, and

$$\gamma_t = \frac{I(t)}{n + s + sr} \tag{2}$$

Therefore, for $t = 0$:

$$\gamma_0 = \frac{I(0)}{n + s + sr}$$

where $I(0) = s + sr$.

Let $Y(i,t)$ denote the number of sybil peers as identified by peer i at time slot t, and $G(i,t)$ denote the number of honest peers as determined by peer i at time slot t, as

$$G(i,t) = n + s + sr - Y(i,t) \tag{3}$$

If we consider that at time slot t, a truster (peer v) of peer i downloads an infected file from peer q, peer i issues a negative rating message. The rating message contains the name of the downloaded file, the download time, and the ID of peer q. $\Omega(v,t)$ represents the average value of information on peer v, calculated from issued rating messages. $\Omega(v,t)$ is proportional to the amount of information on the P2P network. For example, it could be the number of rating messages issued in the network.

In a P2P network, $I(t)$ and $Y(i,t)$ at time slot $t+1$ can be expressed as

$$\begin{cases} I(t+1) = I(t) + \displaystyle\sum_{i=1}^{n-I(t)} \frac{p(I(t) - Y(i,t))}{n + s + sr - Y(i,t)} \\ Y(i,t+1) = Y(i,t) + \displaystyle\sum_{g=1}^{m} \frac{p(I(t) - Y(i,t))\Omega(v,t)}{n + s + sr - Y(v,t)} \end{cases} \tag{4}$$

where, $0 < Y(i,t) < I(t) < (n + s + sr)$.

It is then easy to see that the efficiency of this mechanism depends on $\Omega(v,t)$. In a conventional trust management scheme, a rating message carries information about a single sybil peer, and $\Omega(v,t)$ is simply represented as Ω. In such a case, $\Omega = 1$ [7]. As discussed below, the proposed framework uses a k-means clustering algorithm and dissemination of rating messages, which carry information about a possible sybil q and its trustees. Hence, the information content in each message is $\Omega = r$, where $r > 1$. When r is large, the clustering scheme is able to identify sybils rapidly and accurately.

The number of compromised peers in a network is evaluated with 200 peers, two sybils, and each sybil creates five sybil identities. The downloading probability p is set to 0.5. The average number of trustees for each peer is set to 10. Figure 1 gives $I(t)$ for different Ωs, where $\Omega = \{0, 1, 10, 60\}$. Here, $\Omega = 0$ means that there is no rating messaging so that peers may not be able to provide information to their trustees, and peers in the network may only collect information from their own experience. As Figure 1 shows, the number of infected peers decreases as Ω increases. To increase Ω, statistics can be collected (and estimated) for longer periods of time. As the figure shows, with $\Omega = 0$, all the malware quickly spreads on all peers. As Ω increases, the spreading of malware not only slows down but it is also contained. Figure 2 shows the number of infected peers versus the value of Ω, which in this case is evaluated till it reaches a value of 60.

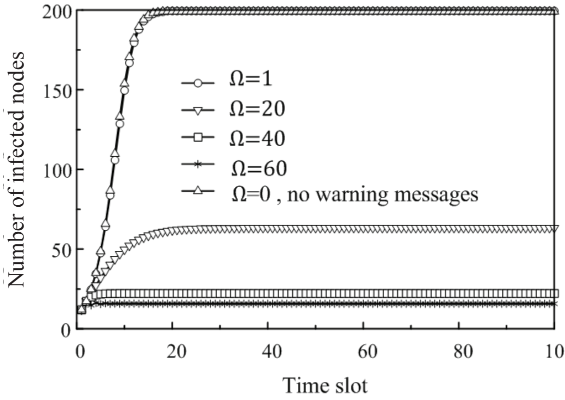

Fig. 1. Number of infected peers under a sybil attack.

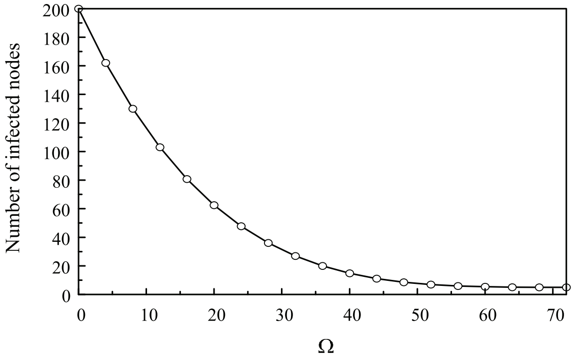

Fig. 2. Number of infected peers for different Ω values.

This analysis shows that an efficient and timely distribution of information about peers and files are necessary and sufficient conditions to contain the effect of a sybil attack on a dynamic trust management scheme. In Section III, we discuss methodologies to increase Ω.

III. Clustering of Sybil Peers and Local Table

In general, honest peers include the estimation of trust value of sybils as these nodes cannot be identified (as sybils). Sybils tend to exercise a strong connectivity among themselves in the attempt to report a large number of interactions. The highly interconnected network may give place to a cluster, identifiable as sybil [20]. We then explore the use of this sybil feature.

In general, a network may be represented as a weighted directed graph $G = (V, E)$, with peers represented by graph V, and trust relationships between a pair of peers $i, j \in V$ represented as directed edges $e \in E$ between i and j. In this paper, $e(i, j)$ means that the edge is directed from peer i to peer j. The direction of the edge determines that peer j is a truster of peer i, and peer i is a trustee of peer j. Edges have different weights, and the weight of $e(i, j)$ is equal to the trust value of peer i about peer j, $T_v(i, j)$. Edges are considered existing only if $e(i, j) > 0$.

A link connecting an honest cluster to a sybil cluster is called *attack edge* [20]. Figure 3 shows some examples. In the figure, let's assume that the central cluster is an honest-peer cluster and that the three clusters around the central cluster are sybil-peer clusters. Sybil clusters I and III have two attack edges. Sybil cluster II has one attack edge. The number of attack edges indicates a larger sybil effect on the network. Therefore, sybils would attempt to increase the number of attack edges. However, building an edge is complex and it may take long time. Therefore, we consider that a feasible number of attack edges that can be set up is rather small.

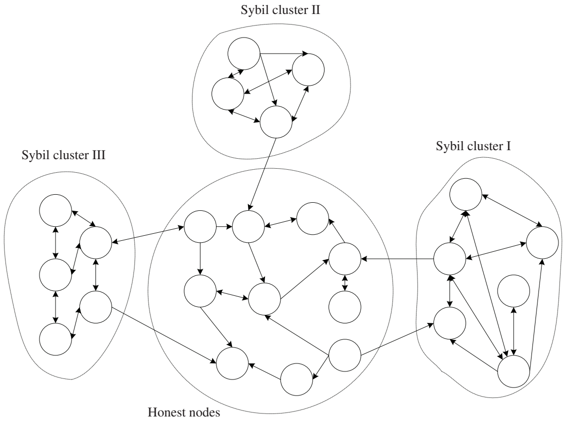

Fig. 3. P2P network with three sybil clusters.

For defining a cluster, let $F(i)$ denote the set of trusters of peer i, and let $FJ(i, j)$ denote the set of peers that are common set trusters of both peer i and j. Therefore, $FJ(i, j) = F(i) \cap F(j)$. Also, $FM(i)$ represent the trusters of peer i as a $1 \times N$ matrix, where N is the total number of nodes in the network, as

$$FM(i) = \{FM(i, 1), \ \ldots, \ FM(i, N)\} \tag{5}$$

where $FM(i, j) = 1$ if peer j is a truster of peer i and $FM(i, j) = 0$, otherwise. The total number of trusters of peer i is

$$TN[F(i)] = \sum_{j=1}^{N} FM(i, j) \tag{6}$$

Figure 4 shows Sybil cluster I, extracted from Figure 3, with six sybils. Let's use the clusters in Figure 4 as an example. The yellow peers in Figure 4(a) are the trustees of peer A, or $F(A)$, and $TN[F(A)] = 4$. The red peers in Figure 4(b) are $F(B)$, and $TN[F(B)] = 4$. Figure 4(c) shows the common trustees of peers A and B, $FJ(A, B)$, and $TN[FJ(A, B)] = 3$. If peer D later claims a set of trustees, and if some of the trustees are common to $FJ(A, B)$ and $TN[FJ(A, B, D)] > TN_{thres}$, D and its peers would be considered sybils with high probability. This example shows that the trusters of sybils are usually a large number of common peers. This is a factor considered by the clustering algorithm in the identification of possible sybils.

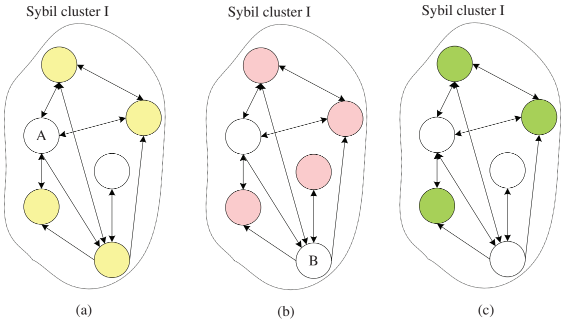

Fig. 4. Trusters' overlap in sybil cluster.

In the proposed framework, each peer also has a local table (i.e., database) to store the list of trusters to be marked as possible sibyl peers. For peer i, peer j is seen as a possible sybil if peer j provides an incomplete download or a file with malware to peer i (or to one of its trusters). $F(j)$ is stored in the local table of peer i, using a format as the example Figure 5 shows. Here, "wins" denotes the size of the time window in which the rating messages are used in the detection of possible sybils. The first row lists the peer IDs. The last two columns, columns j and k, indicate that the message is sent from peer j to peer k. When a rating message is sent from peer j to peer k and $j = i$, it means that there is a download (within a cluster), and this event makes $F(i)$ to be added into the local database.

IV. Verification of Reported Transactions

The verification scheme is based on providing verification of the reported transactions. In this paper, not only the reputation and rating messages are considered but also the set of the trustees of each peer as that may indicate whether the peer is honest or sybil.

Consider that peer i has a public and private keys, namely, $< PK_i, RK_i >$, respectively. It is assumed that public keys are tied to peers using a digital certification authority. Before peer i downloads file f from peer j, peer j sends a transaction guarantee certificate (TG) to peer i. This transaction guarantee $TG(i, j)$ is:

$$TG(j, i) = PK_i\{RK_j\{time,\ f\}\}$$

A rating message sent by peer i is accepted by peer j if and only if $TG(j, i)$ is also provided.

							Sender	Target
Peer index / Time slot	1	2	· · ·	N-2	N-1	N	j	k
t-wins	0	0	· · ·	0.1	0	0	3	5
t-wins+1	0	0.7	· · ·	0.7	0	0.7	N	N-1
t-wins+2	0	0.1	· · ·	0	0	0.1	7	16
t-wins+3	0.2	0	· · ·	0	0	0	N-1	21
⋮	⋮	⋮	⋮ ⋮ ⋮	⋮	⋮	⋮	⋮	⋮
t-4	0.2	0	· · ·	0	0	0.2	6	13
t-3	0	0	· · ·	0.2	0.2	0	7	4
t-2	0	0	· · ·	0	0	0	2	21
t-1	0	0.2	· · ·	0	0	0.2	17	11
t	0.2	0	· · ·	0.2	0	0.2	N	2
Overlap Indicator	1.9	2.7	· · ·	3.9	4.9	5.9		

Fig. 5. Example of local database, with $F(i)$ information.

The objective of this scheme is to generate a proof of an occurred transaction between peers i and j if and only if i and j in fact experienced a transaction with each other. In that case, the transaction proof is sent from peer i to peer j after each download. The transaction proof, $TP(i, j)$, is defined as:

$$TP(i, j) = PK_j\{RK_i\{d, time, f, TG(j, i)\}\} \tag{7}$$

where d is the description of a transaction, as either a satisfying or not satisfying state, namely 0 and 1, respectively. $TG(j, i)$ is embedded in $TP(i, j)$. With TG, peer j is able to prove that it actually uploaded f to peer i at time slot t. Although the use of $TP(i, j)$ is aimed at the verification of transaction, the use of non-existing peers between sybils may still remain.

A. Verification of Reported Interactions for Trust Management

Each peer keeps three tables: $T(i)$, a rating table, and local database. Each peer generates a public/private key and distributes the public key. When a peer looks for a downloading source, the peer makes a decision based on the three tables. It is assumed that file identities remain unchanged for exploitation of file popularity.

The first value in $T(i)$ is T_v, as described before. The second value in the trust table is the propagated value $P_v(i, j)$, and it is calculated from the rating messages generated by the trusters of peer i on peer j. This is:

$$P_v(i, j) = \frac{\text{number of positive propagated ratings}}{\text{total number of propagated ratings}} \tag{8}$$

where $P_v(i, j) \in [0, 1]$. A positive rating is identified by the value of 1, and a negative rating by the value of 0. A small $P_v(i, j)$ means a large probability that peer j is a sybil. The total number of propagated ratings includes both positive and negative ratings.

When peer i seeks f, it chooses a download source from one of the trusters that has f. A conventional scheme selects the truster from a peer whose $T_v(i, j)$ is larger than trust-value threshold t_h as the file source. However, two issues arise: 1) peers need to select a threshold value t_h for the management scheme and 2) the classification of a trustable or untrustable peer may not be accurate sometimes. To correct these problems, the k-means clustering algorithm [21] is used where $T_v(i,j)$ and $P_v(i,j)$ are the x and y axis in a cartesian diagram used for associating peers.

Each peer attempts to identify the cluster of honest peers and the cluster of sybils each time slot. When peer i seeks file f, it sends a request to all the peers in the network, using its private key as follows:

$$RK_i\{i,\ f,\ rt\}. \tag{9}$$

where rt is the request time, which is the time the peer issues the file request. After receiving the request, the peers encrypt the message with the public key of peer i. The honest peers that have f notify peer i and the trustee list and TP associated to the trustee list about it as a rating message. After receiving the search result, peer i eliminates as candidates the peers who are not in its truster list and those peers whose trustees are identified as possible sybils (through the local table).

Peer i then chooses peer j, who has the largest sum of squared values, $T_v^2(i,j) + P_v^2(i,j)$, as the file source. If peer j agrees to be the source, peer j sends a transaction guarantee $TG(j,i)$ to peer i, and the upload begins in the following time slot.

After f is downloaded from peer j, if the download is free of malware and complete, peer i sends a transaction proof TP to peer j that can be used by peer j to prove the contribution to peer i. At the same time, $T_v(i,j)$ is updated and peer i propagates a positive rating message to its trustee set, with the following format:

$$RK_i\{ID_i,\ ID_j,\ \tau,\ TG(j,i)\} \tag{10}$$

where τ is the satisfaction level. Peer k receives a propagated rating message from peer i if this peer is a truster of peer k, and then it proceeds to update P_v. The propagated message expires after a specified time. When peer k receives an expired message, peer k ignores the message.

If the download of file f is unsatisfactory, the transaction proof to peer j is suppressed and peer i records peer j into its local table and updates $T_v(i,j)$. Peer i propagates a negative rating message among its trustee list. The format of the negative rating message is as follows:

$$RK_i\{ID_i,\ ID_j,\ TG(j,i)\} \tag{11}$$

V. Performance Evaluation

A P2P network was simulated by using a mesh topology, with 200 active peers. At the beginning, there are five sybils clusters (each contains 11 peers), and each honest peer performs k-means clustering to identify honest peers from possible sybils.

In the simulation, 30% of sybils send rating messages each time slot to raise the reputation of the main sybil and their collaborator (sybil) peers. Because the sybils cannot get transaction proofs TP from honest peers, each sybil randomly selects six other sybils as their trustees and sends artificial rating messages to them.

The efficiency of the proposed framework is measured in terms of the number of uncompromised peers; or good nodes. Figure 6 shows the simulation results, under a large attack, rate of 0.8. The attack rate is the probability that the sybil peers develop attacks, by issuing artificial rating messages in a time slot. In this figure, K indicates the use of the k-means clustering algorithm, T indicates the use of transaction verification, and L indicates the use of a local table. The figure shows seven curves, each curve represents a different combination of the different mechanisms. The figure shows the effect of using T, K, and L separately. From these, it can be observed the transaction verification scheme (T) has the largest positive impact on the efficiency of proposed framework. The reason for that is because the effect of the local table and and k-means clustering schemes are used to determine sybil candidates but they don't guarantee the absence of false negatives. However, these two mechanisms reduce the probability of selecting a sybil as a downloading source. On the other hand, the transaction verification mechanism is a direct countermeasure to the attack on the reputation of peers, and therefore, this mechanism has the largest countermeasure effect.

This figure also shows that when these three mechanisms are combined, the efficiency is very high. The results show that the proposed framework is able to suppress the proliferation of malware under sybil attacks efficiently, as the number of good nodes (i.e., peers clear of malware) remains at 184. This is, the number of compromised peers is small and it remains at this value indefinitely.

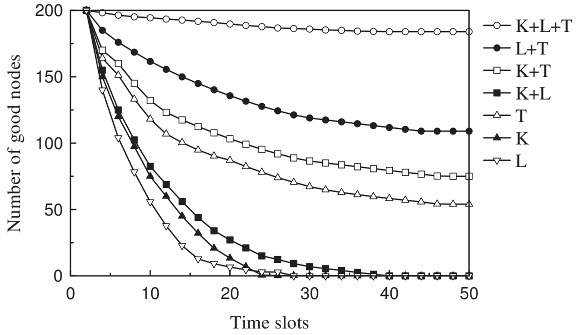

Fig. 6. Comparison of different mechanisms under attack rate 0.8.

Figure 7 shows the same scenarios discussed in Figure 6, under, however, a small rate attack of 0.2. This Figure shows a larger effect of the proposed mechanisms.

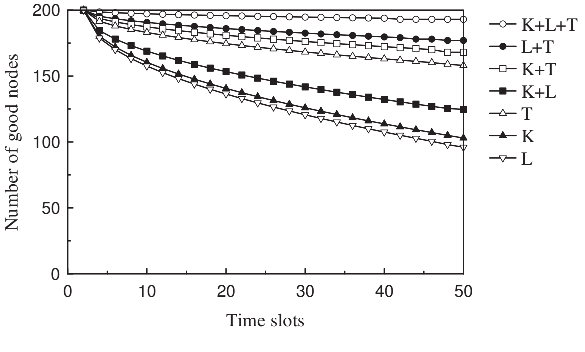

Fig. 7. Comparison of different mechanisms under attack rate of 0.2.

Figure 8 shows the failure ratio of the k-means clustering methods under local table (L), the transaction verification mechanism (T), and the combination of the two ($L+T$). The failure ratio is calculated as the number of sybil peers in the truster list divided by the total number of trusters. This figure shows that the use of the local table and transaction verification mechanisms, the ratio of false positives is reduced.

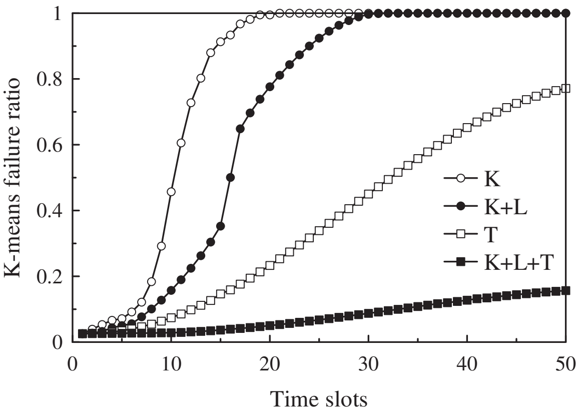

Fig. 8. K-means clustering failure ratio.

VI. Conclusions

Trust management is a strategy to determine the reputation of peers by evaluating the level of trustability. However, a trust management system may not be effective under sybil attacks, as these attacks attempt to increase the reputation of sybil peers and therefore, making them attractive downloading sources for honest peers. To countermeasure that, we proposed a framework that consists of three mechanisms, a local table to determine the collaborators of sybils and honest peers, a k-means mechanism to cluster peers as possible sybils or honest peers, and a transaction verification mechanism to verify that the reported transaction actually occurred. These three mechanisms act on different properties of a sybil attack. We performed a simulation of a P2P network using a trust management scheme under a sybil attack and measured the number of compromised peers. The results show the effects of each mechanism and their combinations. We observed that the transaction verification mechanism is the most effective single strategy as it targets one exploit of sybils, which is the report of fictitious interactions. Furthermore, the results show that the combination of the three mechanisms has the largest effect on supporting trust management as the number of peers compromised is bound to a small number.

The derivation of these three mechanisms origins from our proposed model of a trust management system and information about the probability of a peer of being a sybil. The model is based on a probabilistic approach. The evaluation of the model indicates that it is not sufficient to identify fictitious interactions, but also it is also important to identify (with high probability) potential sybils.

References

[1] G. Theodorakopoulos and J. S. Baras, "On trust models and trust evaluation metrics for ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 2, pp. 318–328, 2006.

[2] L. Cai and R. Rojas-Cessa, "Mitigation of malware proliferation in p2p networks using double-layer dynamic trust (ddt) management scheme," in *Sarnoff Symposium, 2009. SARNOFF'09. IEEE.* IEEE, 2009, pp. 1–5.

[3] P. Herrmann, "Trust-based procurement support for software components," in *Proceedings of the 4th International Conference on Electronic Commerce Research (ICECR-4), Dallas, ATSMA, IFIP*, 2001, pp. 505–514.

[4] Y. Wang and J. Vassileva, "Trust and reputation model in peer-to-peer networks," in *Peer-to-Peer Computing, 2003.(P2P 2003). Proceedings. Third International Conference on.* IEEE, 2003, pp. 150–157.

[5] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *Proceedings of the 12th international conference on World Wide Web.* ACM, 2003, pp. 640–651.

[6] S. Marti and H. Garcia-Molina, "Limited reputation sharing in p2p systems," in *Proceedings of the 5th ACM conference on Electronic commerce.* ACM, 2004, pp. 91–101.

[7] X. Ding, W. Yu, and Y. Pan, "A dynamic trust management scheme to mitigate malware proliferation in p2p networks," in *Communications, 2008. ICC'08. IEEE International Conference on.* IEEE, 2008, pp. 1605–1609.

[8] E. Damiani, D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante, "A reputation-based approach for choosing reliable resources in peer-to-peer networks," in *Proceedings of the 9th ACM conference on Computer and communications security.* ACM, 2002, pp. 207–216.

[9] J. Shin, T. Kim, and S. Tak, "A reputation management scheme improving the trustworthiness of p2p networks," in *Convergence and Hybrid Information Technology, 2008. ICHIT'08. International Conference on.* IEEE, 2008, pp. 92–97.

[10] L. Cai and R. Rojas-Cessa, "Bounding virus proliferation in p2p networks with a diverse-parameter trust management scheme," *Communications Letters, IEEE*, vol. 13, no. 10, pp. 812–814, 2009.

[11] ——, "Three-dimensional based trust management scheme for virus control in p2p networks," in *Communications (ICC), 2010 IEEE International Conference on.* IEEE, 2010, pp. 1–5.

[12] D. Quercia and S. Hailes, "Sybil attacks against mobile users: friends and foes to the rescue," in *INFOCOM, 2010 Proceedings IEEE.* IEEE, 2010, pp. 1–5.

[13] B. Xiao, B. Yu, and C. Gao, "Detection and localization of sybil nodes in vanets," in *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks.* ACM, 2006, pp. 1–8.

[14] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," *ACM Computing Surveys (CSUR)*, vol. 42, no. 1, p. 1, 2009.

[15] G. Guette and B. Ducourthial, "On the sybil attack detection in vanet," in *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on.* IEEE, 2007, pp. 1–6.

[16] T. Zhou, R. R. Choudhury, P. Ning, and K. Chakrabarty, "Privacy-preserving detection of sybil attacks in vehicular ad hoc networks," in *Mobile and Ubiquitous Systems: Networking & Services, 2007. MobiQuitous 2007. Fourth Annual International Conference on.* IEEE, 2007, pp. 1–8.

[17] N. B. Margolin and B. N. Levine, "Quantifying resistance to the sybil attack," in *Financial Cryptography and Data Security.* Springer, 2008, pp. 1–15.

[18] M. Srivatsa, L. Xiong, and L. Liu, "Trustguard: countering vulnerabilities in reputation management for decentralized overlay networks," in *Proceedings of the 14th international conference on World Wide Web.* ACM, 2005, pp. 422–431.

[19] A. Cheng and E. Friedman, "Sybilproof reputation mechanisms," in *Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems.* ACM, 2005, pp. 128–132.

[20] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "Sybilguard: defending against sybil attacks via social networks," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 267–278, 2006.

[21] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 281-297. California, USA, 1967, p. 14.