Personalizing EigenTrust in the Face of Communities and Centrality Attack

Conference Paper · March 2012

DOI: 10.1109/AINA.2012.48

CITATIONS
17

READS
126

4 authors, including:

Nazareno Andrade
Universidade Federal de Campina Grande (UFCG)
56 PUBLICATIONS **1,434** CITATIONS

SEE PROFILE

Dimitra Gkorou
ASML
14 PUBLICATIONS **62** CITATIONS

SEE PROFILE

J.A. Pouwelse
Delft University of Technology
182 PUBLICATIONS **3,789** CITATIONS

SEE PROFILE

Personalizing EigenTrust in the face of Communities and Centrality Attack

Nitin Chiluka*, Nazareno Andrade†, Dimitra Gkorou*, Johan Pouwelse*
*Delft University of Technology, the Netherlands
†Universidade Federal de Campina Grande, Brazil
Email: n.j.chiluka@tudelft.nl

***Abstract*—EigenTrust (ET) is a renowned algorithm for reputation management in adversarial P2P systems. It incorporates the opinions of all peers in the network to compute a *global* trust score for each peer based on its past behavior, and relies on a set of *pre-trusted nodes* to guarantee that malicious nodes cannot subvert the system. In this paper, we show that ET is vulnerable to community structure and a novel targeted attack based on *eigenvector centrality*, since ET ranks nodes close to the pre-trusted ones higher than those further away.**

To address these shortcomings, we propose Personalized EigenTrust (PET) which (i) enables each user to choose her trusted peers from the social network of peers, thereby eliminating the need of pre-trusted nodes and making the system autonomous, (ii) is effective in networks operating under various transaction models based on distributions such as random, community-like and power-law, and (iii) is robust to many types of attacks including the targeted one based on eigenvector centrality. Our simulation results reveal that PET outperforms ET under diverse transaction models and attack strategies.

I. Introduction

Defending against malicious behavior in open systems is a longstanding challenge. Over the last decade, a family of random walk-based trust schemes have been proposed to counter adversaries in various fields such as web search [9], P2P reputation systems [12], [18], and Sybil defenses [22], [21], [6]. Despite considerable differences, each of these schemes employs a fundamental principle – *random walk with restart from trusted nodes* – to rank nodes in the system in order to differentiate honest nodes from malicious ones. Although very popular, several of these schemes lack a deeper understanding of their robustness. In particular, a recent study [20] shows that Sybil defense schemes (e.g. [22], [21], [6]), a subset in this family, are highly vulnerable to community structure and targeted attacks. In light of this study, we hypothesize that these concerns may affect much broader range of schemes in this family. In this paper, we revisit a renowned trust scheme in this family – EigenTrust [12] – to verify this hypothesis.

EigenTrust (ET) is an algorithm for reputation management in adversarial P2P systems [12]. It calculates a 'global' trust score for each peer based on its past behavior by incorporating the opinions of all peers in the system. ET relies on a set of *pre-trusted peers* (usually chosen by the system designers) to guarantee that adversaries cannot subvert the system.

Our analysis of ET reveals that peers that are close and better connected to the pre-trusted peers are ranked higher (based on their trust scores) than the rest of the network. This insight has two main implications. First, ET can essentially be viewed as a *community detection* algorithm which identifies communities around pre-trusted peers. As a consequence, peers in communities that are 'far away' from pre-trusted ones would be ranked very low despite potentially being honest.

Second, our insight motivates malicious peers to employ a novel attack strategy based on *eigenvector centrality* [2]. That is, malicious peers improve their ranking by behaving well with top ranked ones, and acting poorly with low ranked peers with little consequence. Since ET is a global ranking algorithm, such a targeted attack has system-wide negative impact. Both these implications highlight two main bottlenecks of ET: *pre-trusted peers* and *global ranking*.

To address these shortcomings, we propose Personalized EigenTrust (PET) which enables each peer to (i) choose its own set of trusted peers by leveraging the trust inherent in the *social network* among peers, and (ii) calculate *personalized rankings* of other peers. PET simply eliminates the need for the pre-trusted ones, thereby making the system *autonomous* in that there is no 'central' element such as pre-trusted peers. As a result, there is no 'single point of attack' that adversaries can predominantly target.

We evaluate the performance of PET under diverse conditions through simulations. We reach the following conclusions. First, PET outperforms ET (i) under various transaction models based on distributions such as random, community-like, and power-law, and (ii) in the face of different attacks such as collusion with spies, centrality-based and traitors. Second, the eigenvector centrality attack is more devastating than any of the attacks previously studied [12], with ET performing worse than a random technique when this strategic attack is performed. Third, ET performs particularly poorly under community-based transaction models. Lastly, but equally as important, social network-based trusted peers in PET address the *cold start* problem of newcomers during their bootstrap phase into the system. This potentially improves user experience which is vital in retaining the newcomers in the system.

The insights from our study have important implications for existing and future designs of random walk-based trust schemes in adversarial systems. First, such trust schemes need to be evaluated under diverse transaction models to better understand their strengths and weaknesses. Second, trust schemes based on (variants of) pagerank or max-flow need to be evaluated under attacks based on eigenvector- or

betweenness centrality, respectively.

Our study is significantly different from previous efforts at personalization of ET [4], [10], [5]. At a high level, their main premise is that the pre-trusted peers could either leave the system or be compromised, thereby weakening the system. Although their concerns are valid, our study shows that, even in a churn-free and non-compromised system, ET fares poorly when the network exhibits community structure and/or centrality attacks are performed.

The rest of the paper is organized as follows. In section II, we analyze ET critically and identify its vulnerabilities. In section III, we propose our approach to address the challenges in ET. Next we evaluate our approach under wide range of scenarios in sections IV, V and VI. Section VII reviews the related work, while section VIII presents concluding remarks.

II. Analysis of EigenTrust

Our goal in this section is to better understand the process by which EigenTrust [12] computes the rankings of peers.

A. Algorithmic overview

Consider a P2P file-sharing system in which a peer i after downloading a file from another peer j rates the transaction as positive $(+1)$ or negative (-1) based on whether the downloaded file was authentic or not. Peer i then computes a *local trust value*[1] S_{ij} which represents its net number of authentic downloads from peer j. Next, peer i computes a *normalized local trust value* $C_{ij} = max(S_{ij}, 0)/\sum_j max(S_{ij}, 0)$ representing the extent of trust i has in j ($C_{i,j} \in [0, 1]$). Since a typical peer i downloads from only a small subset of the population, its trust-based view $C_{i,*}$ of the network is limited.

To expand its knowledge, peer i incorporates the opinions of its neighbors and the neighbors of its neighbors and so on to compute its reputation vector $\vec{r}_i = (C^T)^n C_{i,*}$ after n iterations. For large n, each peer's $\vec{r}_i$ converges to the same vector $\vec{r}$, the left principal eigenvector of C. In essence, $\vec{r}$ is considered the *global reputation vector*, and $\vec{r}[j]$ denotes the *global reputation score* of peer j.

To counter the gaming of the system, ET employs a set of *pre-trusted peers* which all peers (or at least the honest ones) trust such as the designers and early users, since they are less likely to disrupt the system. The opinions of these pre-trusted peers are weighed more than the others to compute the final global reputation vector $\vec{r}$, which is redefined as:

$$\vec{r} = \alpha C^T \vec{r} + (1 - \alpha)\vec{d} \tag{1}$$

where the teleportation parameter $\alpha \in [0, 1]$ is a constant, and $\vec{d}$ is the static distribution vector representing the set of pre-trusted peers. For every pre-trusted peer i, $\vec{d}[i] = 1/p$ where p is the number of pre-trusted peers; otherwise, $\vec{d}[i] = 0$.

ET can also be interpreted in the context of Markov chains. Let the trust matrix C be modeled as a weighted directed graph in which a peer represents a node and the element $C_{i,j}$ represents the weight of the edge from peer i to peer j. Consider a random walker, starting from one of the pre-trusted peers, wandering the graph. When at a particular node i, this walker with a probability α selects a random neighbor j based on $C_{i,j}$, and with a probability $1 - \alpha$ jumps to one of the pre-trusted peers. The more often this walker visits a node, the higher its reputation score (and hence, its ranking).

[1]*Notations:* Given a matrix A, let vectors $A_{i,*}$ and $A_{*,j}$ denote the i-th row and the j-th column of the matrix respectively, and $A_{i,j}$ be the element of their intersection. Given a vector $\vec{a}$, let $\vec{a}[i]$ represent its i-th element.

B. ET as a community detection algorithm

Since each random walk starts from one of the pre-trusted nodes, we hypothesize that nodes close to the pre-trusted nodes have a higher likelihood to be visited by the random walker, and hence, have higher reputation scores compared to farther nodes. To verify our hypothesis, we adopt a similar methodology as [20]. Specifically, we first build a synthetic network using Barabasi-Albert preferential attachment model [1] with 512 nodes and an initial node degree of 8, and then rewire it such that two densely connected communities ($G1$ and $G2$) of 256 nodes each are formed, connected by a small number of edges (Fig. 1(a)). Next, we randomly choose a node in one of the communities (say $G1$) as the pre-trusted node. We then compute ET reputation scores for all nodes in the network, based on which we calculate their rankings.

Figure 1(b) illustrates how closely the top-k ranked nodes are connected to the pre-trusted node. The horizontal axis denotes the size of the partition containing top-k ranked nodes. For example, the value of $k = 10$ splits the ranking into two parts: one with the top-10 ranked nodes, and the other with the rest. The vertical axis denotes the *conductance* of the partition containing top-k ranked nodes. Conductance is a widely used metric to evaluate the quality of community structure [13] which measures how closely a subset of nodes are connected among themselves relative to the rest of the network. The values of conductance range from 0 to 1, with smaller values indicating stronger communities.

Figure 1(b) plots the conductance as we vary the size of the partition containing the top-k ranked nodes. Most of the nodes in the same community ($G1$) as the pre-trusted node are ranked higher than those in the other community ($G2$). This is because the former are better connected to the pre-trusted nodes than the latter. There is also a strong community structure for the partition of nearly half the nodes in the network. However, adding more nodes from the bottom half ranked peers increases conductance, thereby decreasing the community structure for larger partitions.

Hence, ET can be viewed as a *community detection* algorithm which identifies communities around pre-trusted nodes.

C. Eigenvector centrality attack & its impact

ET's trust score is essentially a variation of *eigenvector centrality* [2] which is a measure of the relative importance of a node in a graph. The global reputation score, and thus the rank, of a peer in ET can be viewed as a measure of its eigenvector centrality with respect to pre-trusted peers. Rephrasing the finding in Sec. II-B, nodes close to the pre-trusted peers are top (eigenvector) central peers, while the nodes further away

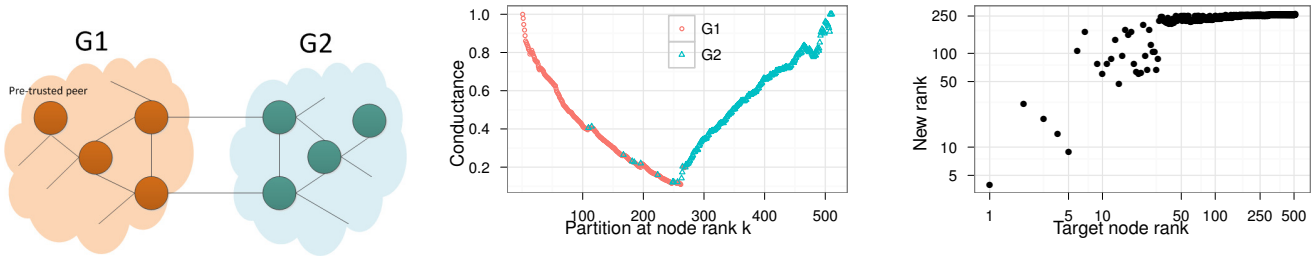

Fig. 1. (a) (*left*) A pictorial illustration of a network with two communities $G1$ and $G2$, with the pre-trusted node in $G1$. Fig 1(b) (*middle*) Conductance versus partitions based on node rank. Color represents the community to which the node with a rank k belongs. Conductance shows the quality of community structure of top-k ranked nodes relative to the rest of the network. Fig 1(c) (*right*) New rank obtained by targeting a node with a rank k.

are less central peers. Hence, a straightforward attack strategy for malicious peers is to form edges with top central peers in order to get close to the pre-trusted peers, thereby gaining high rank which they then exploit to act poorly with the rest of the peers (which are less central).

We now validate this attack strategy by demonstrating how an adversary can improve its rank in the system. We first construct a synthetic network using Barabasi-Albert preferential attachment model [1] with 512 nodes and an initial degree of 8, and then we randomly choose a pre-trusted node. Next, we choose a node i with an average rank (here 256), and add one edge to another target node j.

Figure 1(c) plots the new rank of this node i versus the rank of the target node j. Targeting top ranked nodes significantly improves i's ranking. For instance, just a single edge with a top-5 ranked node increases i's ranking from 256 to below 50. However, adding an edge to a lower ranked node has little effect on i's ranking. This confirms that a strong strategy for malicious nodes is to behave well with top central nodes, and to act maliciously with the rest with little consequence.

D. Limitations

The insights gained from Sec. II-B and Sec. II-C have negative implications on two key aspects of ET:

1) Pre-trusted peers: The coverage of random walks from pre-trusted peers becomes vital to the performance of ET. As the system grows, more and more communities typically evolve due to varied tastes in users. If the pre-trusted peers are selected poorly, nodes in *far off* communities would be ranked very low despite being honest.

2) Global ranking: Since ET is a global ranking mechanism, a centrality attack will have system-wide negative impact. For instance, strategic malicious peers can behave well with a few top ranked peers (minority), while behaving maliciously with the others (majority). In addition, the rating by a user is often subjective, and hence a general agreement is difficult to reach in all scenarios. Consider a typical example: a poor quality video may be disregarded by some users, but may suffice for others due to its content and significance.

In short, global consensus on pre-trusted nodes makes ET ineffective when either the network exhibits community structure or centrality attack is applied.

III. Personalized EigenTrust

We argued in the previous section that the two main bottlenecks of ET are *pre-trusted peers* and *global ranking*. Here we describe our approach to address these shortcomings.

A. Approach

We propose Personalized EigenTrust (PET) which enables each peer to (i) choose its own set of trusted peers, instead of the pre-trusted ones as in ET, and (ii) formulate its personalized view of the system, instead of a global one as in ET. More formally, a peer i computes its *personalized reputation vector* $\vec{r_i}$ (similar to Eq. (1)) using its set of trusted peers (including itself) in the form of *trust vector* $\vec{d_i}$:

$$\vec{r_i} = \alpha C^T \vec{r_i} + (1-\alpha)\vec{d_i} \tag{2}$$

It is possible to interpret PET in two different perspectives. From the centrality point of view, the personalized reputation score of a peer j from another peer i's perspective is a measure of j's eigenvector centrality with respect to i's trusted peers (which includes itself). From a random walker perspective, a peer i wishing to evaluate other peers in the system begins a random walk from one of its trusted peers (including itself) and then wanders the system. With a probability $1-\alpha$, this walker jumps back to one of its trusted peers and restarts the random walk. This is significant on two fronts. First, the walker traverses mostly the relevant part of the system which gives her a personalized view. Secondly, the walker remains close to its trusted nodes, and hence reduces the chances of escaping into the malicious region.

PET leverages the trust inherent in the likely existent social network among the peers, either online or offline. The use of social network-based trusted peers eliminates the need for the pre-trusted peers. Essentially, this makes the system *autonomous* in that there is no 'central' element such as pre-trusted peers that significantly influences reputation evaluation. As a result, there is no 'single point-of-attack' that adversaries can predominantly target. Another main advantage of being autonomous is that the system becomes self-sufficient and self-organizing, which are often highly desired properties in a P2P network. We discuss more details on the role of social network in Section VI.

B. Basic computation

We now describe how each peer i computes its personalized reputation vector $\vec{r_i}$. Peer i first gathers all columns of the matrix C which simply correspond to the respective peers. Next, i computes $\vec{r_i}$ using its trusted peers in the form of $\vec{d_i}$ (Eq. (2)). We note that the information of its trusted peers is kept locally. This ensures that the personalized reputation vector $\vec{r_i}$ is *private* to i. Here we assume that the computation of PET is oblivious to the underlying overlay network. That is, the matrix C can be gathered using gossiping in unstructured networks or DHTs in structured networks.

We now discuss the complexity of computing PET in this fashion. Suppose there are n peers in the network, and their transactions result in the matrix C having $m = \mathcal{O}(n)$ non-zero elements. Here, we assume that C is sparse, i.e., $m << n^2$. For each peer i, computing its personalized reputation vector $\vec{r_i}$ takes $\mathcal{O}(m)$ network overhead, $\mathcal{O}(m)$ storage space, and $\mathcal{O}(ml)$ time where l is the number of iterations for $\vec{r_i}$ to converge. This implies that, for a network of 1 million nodes and 10 million non-zero floating point elements in C, the storage required and the network overhead for each peer is around 40 MB. Considering the fact that the size of today's popular files in P2P networks is in the order of 1 GB, downloading 40 MB required for reputation calculation seems reasonable since the goal in adversarial P2P file-sharing systems is to download an authentic version of a file the very first time. This improves user experience as well as reduces the wastage of resources in terms of time and bandwidth.

C. Scalable computation

We now describe how PET can be computed in a scalable fashion. Let R be a matrix in which the i-th column represents the vector $\vec{r_i}$ (=$R_{*,i}$). Also, let D be defined similarly. Bringing all under one umbrella, Eq. (2) is reformulated as:

$$R = \alpha C^T R + (1 - \alpha)D \tag{3}$$

Solving for R, Eq. (3) is rewritten as:

$$R = (1 - \alpha)(I - \alpha C^T)^{-1} D \tag{4}$$

The right-hand side of Eq. (4) comprises two parts. The first part $(I - \alpha C^T)^{-1}$ takes into account the transaction behavior of peers in the system, while the second part D represents the trust relationships in the social network. This allows us to clearly differentiate the computational aspects as well.

The matrix inverse $S = (I - \alpha C^T)^{-1}$ can be efficiently computed using matrix dimensionality reduction techniques such as proximity embedding [19]. This technique factorizes S into two matrices U and V of smaller dimensions such that $S_{n \times n} = U_{n \times p} V_{p \times n}$ where $p << n$. We employ a *super-peer approach* to compute U and V by exploiting the heterogeneity among peers in the network in terms of resources such as computational power, memory and online time [3]. A fraction of powerful peers are elected as *score managers* which coordinate to compute U and V in a distributed fashion. Song et al [19] have observed that the factorization of S was nearly 15 times faster by using 150 peers compared to a single instance.

To compute $R_{*,i}$, peer i first requests the vector $S_{*,j}$ for each of i's trusted peers j from one of the score managers, and then adds them: $R_{*,i} = (1 - \alpha) \sum_j S_{*,j} D_{j,i}$. We note here that the answering score manager computes $S_{*,j} = UV_{*,j}$. Alternatively, i could request only a subset of rows of $S_{*,j}$ instead of the entire vector. For instance, when a query returns hits from a set of peers L, i can request only the rows of $S_{*,j}$ corresponding to peers in L.

IV. Simulation Setup

We build our simulation model along the lines of the original ET study[2]. At a high level, the model represents a P2P file-sharing system consisting of *honest* as well as *malicious* peers downloading files from each other. Honest peers in such an adversarial network would want to download only *authentic* files. Conversely, malicious peers would want honest peers to download *inauthentic* files. To do so, malicious peers can resort to various attacks that potentially undermine the usefulness of the system, particularly to honest peers. To defend against such attacks, honest peers employ a trust-based algorithm to evaluate the trustworthiness of other peers. In the rest of the paper, we use the percentage of inauthentic downloads as the basic metric to evaluate the performance of PET against the baselines ET and Non-Trusted (NT).

We now describe the simulation execution. An experiment proceeds in simulation cycles which in turn is divided into query cycles. During a query cycle, each honest peer i in the network may either issue a query or be passive. When a query is issued, the peer is returned with hits from a subset of all peers in the network. The ratio of good and bad hits among them is equal to the proportion of honest and malicious peers in the network. The peer i then selects one peer among these hits, based on latter's trustworthiness, to download the file. Peer i then rates the transaction as positive (+1) or negative (-1) based on whether the downloaded file was authentic or not. At the end of each simulation cycle, a trust-based algorithm is used to compute reputation scores which are later used in the subsequent simulation cycle for download source selection. Also, statistics such as the number of authentic and inauthentic downloads by each peer are collected. Each experiment is run multiple times over which the results are averaged.

As Section II-B has highlighted the role of the topology generated by the transaction patterns of peers, we also consider this factor in our experimental setup. For that, we evaluate how PET performs in diverse transaction models based on following distributions:

- *Random*: Content is randomly distributed among peers, and the queries answered by a peer are also random.
- *Community*: There are multiple communities in the system, and a peer usually answers queries issued by peers in the same community.
- *Power-law*: A few peers among the entire population have the most popular content, and answer majority of the

[2]We refer the reader to [12] for more details.

TABLE I
SIMULATION SETTINGS

Parameter	Default/Range
# Total nodes	1000
# Pre-trusted nodes	10
% Malicious peers	50%
Transaction model	random, community, power-law
% peers answering a query	5%
Communities	G1 (75%) and G2 (25%)
Power-law exponent γ	0.8 ([0.0,2.0])
Teleportation parameter α	0.15
# Simulation cycles	30
# Query cycles	50
# Runs for each experiment	5

queries issued in the network. The probability that a peer i answers a query is proportional to $i^{-\gamma}$.

In each experiment in the next section, we simulate a network of 1000 peers comprising the same number of honest and malicious peers. Each query by an honest peer is returned with hits from 25 honest peers and 25 malicious peers (5% of the population). While the subset of honest peers answering a query is based on a particular transaction model, the subset of malicious peers answering the query is always based on power-law distribution where $\gamma = 1$. This enables malicious peers to gain a vantage point against honest peers. Table I shows various parameters and their default values or ranges used in the simulation.

V. RESILIENCE TO ATTACKS

In this section, we evaluate the performance of PET under various attacks by considering that each peer trusts only itself. In other words, each peer i computes its personalized reputation vector $\vec{r_i}$ by assigning $\vec{d_i}[i] = 1$. In the next section, we discuss the role of social network-based trusted peers.

Due to space limitations, we omit the results of Threat A, B and C proposed in the ET paper [12]. Harm caused by each of these threats is less than Threat D which we discuss next.

A. Collusion with malicious spies

A subset of malicious peers acting as *spies* upload authentic files when selected as download source. At the same time, these spies assign high trust scores to the rest of malicious peers which always upload inauthentic files as well as form a malicious collective / ring among themselves.

Figure 2 plots the percentage of inauthentic downloads versus the percentage of spies among malicious peers. PET outperforms ET in various scenarios. We particularly note that ET performs poorly in the community model. This is because peers in the community with no pre-trusted peers have low reputation scores, and hence differentiating them with low ranked malicious peers is difficult. However, both PET and ET perform very well in the power-law model, since only a few top ranked honest peers answer most of the queries.

B. Eigenvector centrality attack

We now reformulate the eigenvector centrality-based attack, discussed in Sec. II-C, to suit the current context of simulation environment. That is, malicious peers upload authentic files to topmost eigenvector central peers, while uploading inauthentic files to the rest of honest peers. At the same time, these malicious peers collude to form a ring among themselves.

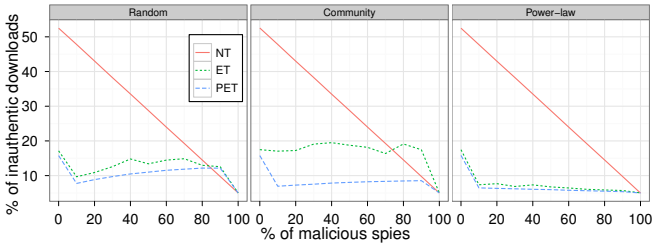

Fig. 2. Impact of colluding spies (Threat D).

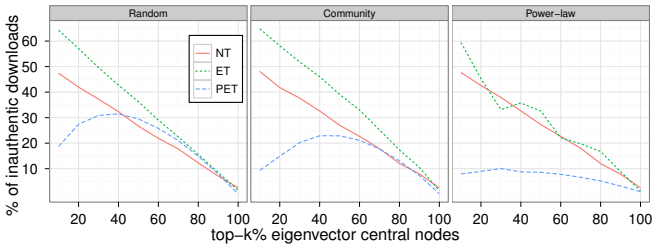

Fig. 3. Impact of eigenvector centrality attack.

Since the notion of centrality of a peer j in PET is different from the perspective of each peer i in the network, the strategy for malicious peers is non-trivial (unlike in ET). Malicious peers assign the values in the eigenvector of $(I - \alpha C^T)^{-1}$ to the corresponding peers, which are fundamentally their eigenvector centrality scores. The intuition behind this idea is that the globally topmost central nodes are also highly central from many peers' individual perspective in the context of PET.

We now examine the impact of such an attack. Figure 3 plots the percentage of inauthentic files downloaded versus the percentage of topmost central nodes to which strategic malicious peers upload authentic files. We make the following observations. First, the impact of this attack is significantly stronger than of any of the previous attacks studied [12]. For instance, in the random model, the performance of PET under this attack is nearly three times worse than that under Threat D. Second, PET outperforms ET in various scenarios. This result highlights the difficulty in attacking a personalized reputation system as compared to a global one. Third, for the random and community models, the performance of PET gradually decreases until malicious peers behave well with less than top 40% central nodes. However, the performance of PET improves when more central nodes are targeted but remains almost as bad as NT's. Lastly, ET performs worse than NT in most cases. This is because many malicious peers gain a lot higher reputations than their honest counterparts, and hence they are often selected for download.

We use the centrality-based attack strategy for malicious peers in the experiments in the rest of the paper.

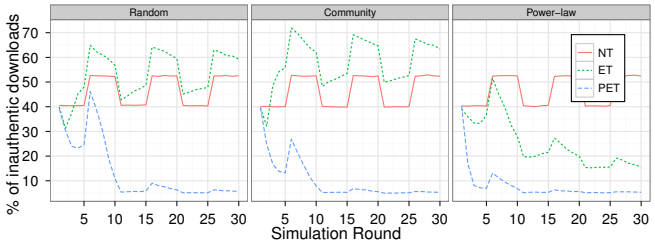

Fig. 4. Impact of traitor attack.

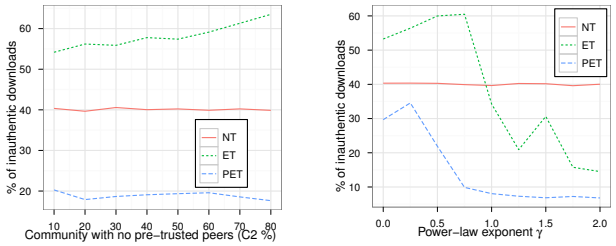

Fig. 5. Impact on transaction models. 5(a) Community. 5(b) Power-law.

C. Extended Traitor attack

In this attack, malicious peers 'milk' the reputation in that they alternate between (i) uploading authentic files for some time to gain reputation, and (ii) later uploading inauthentic files to harm the system. We simulate this attack by extending the centrality-based attack. For five simulation cycles, malicious peers upload authentic files only to the top-25% central nodes. For the next five simulation cycles, malicious peers upload inauthentic files whenever selected for download. This process is continued until the end of the experiment.

Figure 4 plots the percentage of inauthentic downloads versus the simulation round. PET outperforms ET in various scenarios. In the case of PET, the traitor attack is counter-productive for malicious peers. The percentage of inauthentic downloads significantly decreases as the time passes since the reputation scores of malicious peers diminish to the extent that they are less likely to be selected for downloading in subsequent simulation cycles. In the case of ET, the reputation scores of a few malicious peers which answer most queries remain so high that they lose little despite acting maliciously for a few rounds in random and community models. However, in the power-law model, since a few honest and malicious peers which answer most queries compete with each other, the reputation scores of malicious peers reduce well below those of honest peers when acting maliciously for a few rounds.

D. Impact on transaction models

Figure 5(a) plots the percentage of inauthentic downloads versus the percentage of the community with no pre-trusted peers (G2%). The performance of PET is least affected by increasing G2%. This shows that PET can be very effective even for small, new or growing communities. On the other hand, the performance of ET gradually worsens as G2% increases. This is because more peers are 'further away' from the pre-trusted nodes, and hence lesser reputed.

Figure 5(b) plots the percentage of inauthentic downloads versus the power-law exponent γ. Since $\gamma = 0$ is essentially equivalent to the random query model, we note that the performance of both PET and ET significantly improves as the transaction model moves from random to power-law distribution. This is because, in the power-law model, only a few peers answer a majority of the queries. This helps these peers gain higher reputation than malicious peers, thereby increasing the likelihood of getting selected for download.

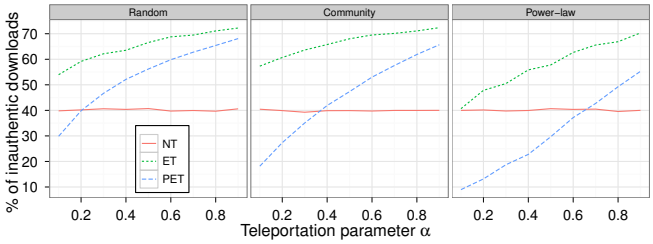

Fig. 6. Sensitivity to teleportation parameter α.

E. Sensitivity to teleportation parameter α

Figure 6 plots the percentage of inauthentic downloads versus the teleportation parameter α. The larger the value of α, the worse the performance of both PET and ET. This shows a long random walk from a trusted peer is more likely to end at a malicious peer than at an honest one. This finding is consistent with that of Whanau [14]. Hence, a short random walk is more suitable in such adversarial systems.

F. Impact on ranking of search results

We now focus on the presentation of search results to users which can potentially influence user experience. When a query returns hits from the network, it is vital to show the user the search results from honest peers ranked higher than those from malicious ones. To measure the accuracy of PET and ET at differentiating honest and malicious peers, we use the metric *Area under Receiver Operating Characteristic (ROC) curve* or A' [8]. This metric represents the probability that an honest peer is ranked higher than a malicious one. $A' = 1$ indicates perfect ranking, while $A' = 0.5$ represents random ranking. We perform the same experiment as mentioned in Section V-B, and present the results from a ranking perspective unlike previous experiments where the focus was on reducing inauthentic downloads by honest peers in the system.

Figure 7 plots the area under ROC curve versus the percentage of topmost central nodes to which strategic malicious peers upload authentic files. The ranking for both PET and ET worsens as we target more top ranked nodes. When more than half the top ranked nodes are targeted, PET performs worse than NT which means that malicious peers are ranked higher than honest peers in the ranking of search results. However, ET's ranking is worse than NT's in each scenario, implying that malicious peers are more likely to be chosen for download.

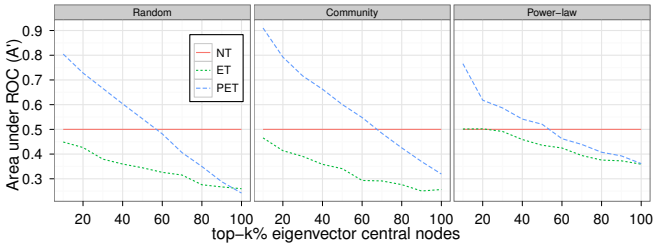

Fig. 7. Quality of ranking of search results.

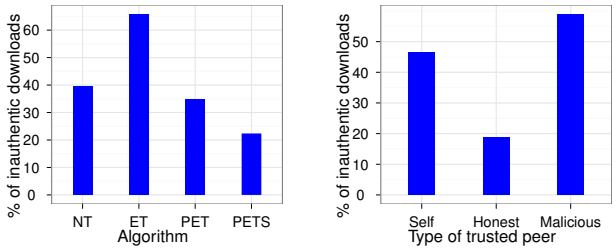

Fig. 8. Robustness during bootstrap. 8(a) Performance of each algorithm for new honest peers. 8(b) Impact of the choice of trusted peers.

This also explains the poor performance of ET in Sec. V-B.

VI. Role of Social Network

We now discuss how social network fits in our PET design.

A. Bootstrapping into the network

1) Invitation: We propose that entry into the network is based solely on *invitations*. That is, a new user needs to get invited by an existing user (typically a friend or a trusted acquaintance) in order to join the network, similar to the initial phases in online social networks such as Orkut and Google+ as well as BitTorrent Darknets. This approach addresses two important challenges. First, when a new user installs a P2P client and runs it for the first time, the client usually contacts either a central server (e.g., Mainline DHT in BitTorrent and Azureus) or a set of fixed super peers (e.g., Tribler, Skype) in order to connect to existing peers in the network. If such a central entity is down or compromised, new clients will not be able to join the network. Using our approach, a new peer obtains contact details of the existing peers from its (trustworthy) inviter. Second, the network gradually evolves around a core of honest peers who also have social relationships among themselves. In this way, our approach leverages the trust inherent in the social network among these peers.

2) Social network-based trusted peers: After joining the network, a new peer faces *cold start* problem in that *whom to trust* in an adversarial system is a fundamental question. Personalized EigenTrust using Social network (PETS) enables each peer to choose its own trusted peers. For simplicity in our experiment, we consider a random transaction model where a new honest peer trusts one existing honest peer uniformly at random. As a result, the former computes the personalized reputation scores from the perspective of the latter. We note here that PET stands for the scenario where each peer trusts only itself, while PETS implies that a new honest peer trusts an existing (older) honest peer.

We simulate the bootstrap scenario with a network comprising 250 honest peers and 500 malicious peers for the first 10 simulation cycles, after which 250 new honest peers join the network. We measure the number of authentic and inauthentic downloads by new honest peers at the end of the experiment.

Figure 8(a) plots the comparison of the performance of each algorithm for new honest peers. PETS outperforms all other approaches including PET. In addition, PET performs nearly as worse as NT, while ET performs the worst. Hence, trusting an older honest peer during bootstrap phase helps in significantly reducing inauthentic downloads. As a result, it potentially improves user experience which is vital in retaining the users (especially the newcomers) in the network.

3) Choice of trusted peers: We now examine how the choice of trusted peers affects the performance for the new honest peers. We use a similar setup as the above experiment, except that the 250 new honest peers are divided equally into three categories based on their type of trusted peer. One-third of these newcomers trust only themselves and another one-third trust in an existing honest peer, while the trusted peers for the rest are all malicious.

Figure 8(b) plots the percentage of inauthentic downloads versus the type of trusted peer by a newcomer. We see that trusting an existing honest peer is better than trusting either a malicious peer or just itself. For new honest peers, this implies that *keep your friends closer and your enemies farther*.

B. Defending against Sybil attacks

Social networks have been leveraged by the research community to counter Sybil attacks [7] in open adversarial systems [22], [21], [6], [16], [14], [17]. Since our design of PET is independent of the underlying overlay network, one can adopt any of these schemes in the system. For instance, in a structured network, Whanau [14] may be employed which leverages social networks to defend against Sybils polluting the indexes of DHTs. In an unstructured network, the decision to incorporate data received during gossiping into the database can be based on a mechanism similar to [17] where each peer maintains a *network of friends* as well as a *network of foes* for suspicious peers.

VII. Related Work

Many previous studies have critically analyzed ET and proposed personalized versions of the algorithm [4], [10], [5]. Chirita et al. [4] argue that a subset of pre-trusted peers can be chosen by peers based on their tastes, instead of a fixed set as in ET. They propose a distributed algorithm similar to Personalized PageRank [11] to compute personalized reputation scores for each peer. Its performance was shown to be similar to ET in their work, whereas our study shows that not only PET outperforms ET under diverse scenarios but also ET performs worse than NT when centrality-based attacks are applied. Jansen et al. [10] discuss the vulnerabilities of ET,

particularly when the pre-trusted peers leave the system or are compromised. They propose Federated EigenTrust which employs elected 'representative nodes' in place of pre-trusted peers. While their concerns are valid, our study shows that, even in a churn-free and non-compromised system, ET fares poorly when the network exhibits community structure. More recently, Choi et al. [5] have presented Personalized EigenTrust which computes the normalized trust scores (refer to Sec. II-A) using beta distribution, which is not our main focus in this paper. We note that all these studies lack extensive analysis on how their algorithms perform under various transaction models and attack strategies.

Evaluation of ET on the network traffic dataset of the Maze P2P file-sharing system by Lian et al. [15] shows that ET produces less than ideal reputation scores for both honest and malicious peers under various collusion patterns. Based on the results of our study (refer to Sec. V-E), we suspect that such poor performance can be explained by their use of very high value of α (=0.9 in their work). The higher the α, the higher the likelihood of a random walk escaping into the malicious region. It will be interesting to see the results of their experiments when run with lower values of α. Here we note that α in our study is equivalent to $1 - \alpha$ in theirs.

VIII. Concluding Remarks

We empirically show that ET is highly vulnerable to community structure and eigenvector centrality attack, and also that its two main bottlenecks are pre-trusted peers and global ranking. To address these shortcomings, we propose Personalized EigenTrust which enables each peer to (i) calculate personalized reputation scores of other peers, and (ii) choose its own set of trusted peers from the *social network* among peers. The use of social network-based trusted peers simply eliminates the need for the pre-trusted ones, thereby making the system *autonomous*. In addition, the social network enables newcomers to bootstrap into the network seamlessly as well as defend against Sybil attacks. Our evaluation on synthetic networks operating under diverse transaction models (random, power-law and community) and targeted attacks including the novel centrality attack reveals that PET outperforms ET in all these scenarios. The insights gained in our study have important implications to existing and future designers of flow-based reputation systems. For instance, pagerank- and maxflow-based trust schemes need to be evaluated under eigenvector- and betweenness-centrality attacks.

The concept of PET is very general and has wide range of applications in P2P transaction networks[3] (Table II) which typically attract malicious behavior, in both centralized and distributed systems. For instance, online content voting systems such as YouTube and Digg face attacks where poor content is rated highly by many adversaries, thereby attaining a space on *Most Popular* pages. To counter such attacks, PET can exploit both voting patterns and social network relationships of users to filter out bad content and uploaders, and recommend only relevant and trustworthy content to users.

[3]Here, 'P2P' refers to relationships such as peer-to-peer, person-to-person and user-to-user.

TABLE II
P2P Transaction Networks: Analogies

File-sharing	Marketplace	Service-oriented	Social media
download	transaction	subscription	view
file	item	service	video/photo/story
uploader	seller	provider	uploader
downloader	buyer	consumer	viewer

Acknowledgment

This work was partially supported by the European Commission's 7th Framework Program through the P2P-Next and QLectives projects (grant no. 216217, 231200).

References

[1] A. L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, pages 509–12, 1999.
[2] P. Bonacich and P. Lloyd. Eigenvector-like measures of centrality for asymmetric relations. *Social Networks*, 2001.
[3] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making gnutella-like p2p systems scalable. In *SIGCOMM '03*.
[4] P. A. Chirita, W. Nejdl, M. T. Schlosser, and O. Scurtu. Personalized reputation management in p2p networks. In *ISWC '04*.
[5] D. Choi, S. Jin, Y. Lee, and Y. Park. Personalized eigentrust with the beta distribution. In *ETRI '10*.
[6] G. Danezis and P. Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *NDSS '09*.
[7] J. R. Douceur. The sybil attack. In *IPTPS '01*.
[8] J. Fogarty, R. S. Baker, and S. E. Hudson. Case studies in the use of roc curve analysis for sensor-based estimates in human computer interaction. In *GI '05*.
[9] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *VLDB '04*.
[10] R. Jansen, T. Kaminski, F. Korsakov, A. S. Croix, and D. Selifonov. A priori trust vulnerabilities in eigentrust. Technical report, University of Minnesota, 2008. http://www-users.cs.umn.edu/~jansen/papers/fet-csci5271.pdf.
[11] G. Jeh and J. Widom. Scaling personalized web search. In *WWW '03*.
[12] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *WWW '03*.
[13] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information networks. In *WWW '08*.
[14] C. Lesniewski-Laas and M. F. Kaashoek. Whanau: a sybil-proof distributed hash table. In *NSDI '10*.
[15] Q. Lian, Z. Zhang, M. Yang, B. Y. Zhao, Y. Dai, and X. Li. An empirical study of collusion behavior in maze p2p file-sharing system. In *ICDCS '07*.
[16] A. Mohaisen, N. Hopper, and Y. Kim. Keep your friends close: Incorporating trust into social network-based sybil defenses. In *IEEE INFOCOM '11*.
[17] D. Quercia and S. Hailes. Sybil attacks against mobile users: friends and foes to the rescue. In *INFOCOM '10*.
[18] A. Rahbar and O. Yang. Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Transactions on Parallel and Distributed Systems*, 2007.
[19] H. H. Song, T. W. Cho, V. Dave, Y. Zhang, and L. Qiu. Scalable proximity estimation and link prediction in online social networks. In *IMC '09*.
[20] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove. An analysis of social network-based sybil defenses. In *SIGCOMM '10*.
[21] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *IEEE Symposium on Security and Privacy '08*.
[22] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. In *SIGCOMM '06*.