EigenTrust++: Attack Resilient Trust Management

Conference Paper · January 2012

DOI: 10.4108/icst.collaboratecom.2012.250420

CITATIONS
14

READS
1,757

4 authors:

Xinxin Fan
Chinese Academy of Sciences（CAS）
28 PUBLICATIONS **226** CITATIONS
SEE PROFILE

Ling Liu
Georgia Institute of Technology
650 PUBLICATIONS **18,450** CITATIONS
SEE PROFILE

Mingchu Li
Dalian University of Technology
228 PUBLICATIONS **1,504** CITATIONS
SEE PROFILE

Zhiyuan Su
University of Dundee
9 PUBLICATIONS **171** CITATIONS
SEE PROFILE

EigenTrust^{++}: Attack Resilient Trust Management

Xinxin Fan*†, Ling Liu†, Mingchu Li*‡ and Zhiyuan Su*†
*School of Software Technology, Dalian University of Technology,Dalian, Liaoning, China, 116620
†College of Computing, Georgia Institute of Technology, Atlanta, Georgia 30332
{xinxinyuanfan@mail.dlut.edu.cn, lingliu@cc.gatech.edu, mingchul@dlut.edu.cn (‡Corresponding author), suzhiyuan2006@gmail.com}

***Abstract*—This paper argues that trust and reputation models should take into account not only direct experiences (local trust) and experiences from the circle of "friends", but also be attack resilient by design in the presence of dishonest feedbacks and sparse network connectivity. We first revisit EigenTrust, one of the most popular reputation systems to date, and identify the inherent vulnerabilities of EigenTrust in terms of its local trust vector, its global aggregation of local trust values, and its eigenvector based reputation propagating model. Then we present EigenTrust^{++}, an attack resilient trust management scheme. EigenTrust^{++} extends the eigenvector based reputation propagating model, the core of EigenTrust, and counters each of vulnerabilities identified with alternative methods that are by design more resilient to dishonest feedbacks and sparse network connectivity under four known attack models. We conduct extensive experimental evaluation on EigenTrust^{++}, and show that EigenTrust^{++} can significantly outperform EigenTrust in terms of both performance and attack resilience in the presence of dishonest feedbacks and sparse network connectivity against four representative attack models.**

I. Introduction

Reputation and trust management continue to play an active role in collaborative computing for a broad range of applications, ranging from cloud computing, decentralized network computing, mobile services to social networks and online communities. A common challenge facing these collaborative systems is how to effectively collaborate in accomplishing tasks while mitigating the attacks (threats) throughout collaboration. This is because participants in most collaborative systems may have little knowledge about other participants with whom there is no prior interaction or transactional experiences. Reputation-based trust management has been used as an effective service selection criterion to evaluate how much one can trust others in these collaborative systems [3], be it cloud service provisioning, mobile commerce and entertainment, or social computing, crowd sourcing and social networks.

Trust is often considered as a personal and subjective measure because it is computed primarily based on a set of personalized or subjective factors or evidence, some of which carry more weight than the others for different entities. Furthermore, an individual's subjective trust can be derived from a combination of personal experience and received referrals. On the other hand, reputation is often considered as a collective measure of trustworthiness based on the referrals or feedback ratings from members in a community [6] based on their transactional experiences or direct interactions. Reputation can be seen as a community-wide trust measure obtained by integrating the personal experiences of many members in the given context within the community. Reputation can be aggregated into a single value that represents what the community as a whole thinks about a particular participant (e.g., a service provider, a mobile node, a social network member) [17]. By collecting, distributing and aggregating the feedbacks about a participant's past behaviors, the reputation trust metrics can help participants to decide whom to trust, encourage trustworthy behavior, and deter participation by those who are unskilled or dishonest [10] [18].

Reputation trust has shown to be beneficial in many eCommerce applications where a large number of participants are involved, be it consumer or producer, and many of them do not have prior interaction or experience with one another. eBay and Amazon are two popular and representative Web services that utilize reputation trust to rank participants [2]. Most of the Reputation systems to date are built on the belief that the efforts of attempting to identify malicious participants that may disturb an operational system by providing inauthentic contents or bad services are more effective than the efforts of attempting to identify inauthentic contents and bad services. This is because malicious participants can easily generate a virtually unbounded quantity of inauthentic contents or bad services if they are not identified, banded from, or constrained in the daily operational system [18] [7].

Trust and reputation management have been an active research area over the last decade. Most of the existing research has developed trust and reputation computational models by utilizing formal methods, such as fuzzy logic theory [13] [11], Bayesian network [15], subjective logic [4] [5], social cognitive methods [1] [8], and game theory [9] [16]. EigenTrust [7] is one of the most popular reputation management models to date. It computes the level of trust that a system places on a participant based on the normalized local trust vector of the participant and its eigenvector, enabling the reputation computation and establishment of a participant through direct experiences and feedbacks as well as indirect experiences obtained through its circle of "friends". We use "friends" of a participant to refer to those other participants with whom this participant has had direct or indirect interaction or transaction relationship. Participants connected by such transactional relationship form a collaboration network or "friendship" network. For each participant, its circle of friends refer to those participants that can be reachable from this participant in the "friendship" network by following the

978-1-936968-36-7 © 2012 ICST
DOI 10.4108/icst.collaboratecom.2012.250420

friendship relationships that it has with its direct and indirect friends.

In this paper we argue that reputation based trust should be attack resilient. By attach resilient, we mean that (i) the reputation of a good participant should stay strong even in the presence of malicious participants, (ii) new participants should be able to built their reputation over time, and (iii) the reputation of a malicious participant should be dropped sharply once he is found to be dishonest or misbehave, for example, by generating inauthentic contents or providing bad services beyond the commonly accepted baseline. We argue that even though EigenTrust is the first trust model that identifies and utilizes the circle of friends as an effective way to handle sparse friendship connections in a large network of participants, it suffers from a number of detrimental vulnerabilities. First, it is vulnerable in the presence of the co-existence of malicious participants with dishonest feedback ratings (referrals) from a small number of normal (good) participants. Second, its iterative computation model based on eigenvector propagates trust uniformly to all its friends, making EigenTrust vulnerable to the sparse networks with some sub-network(s) dominated by malicious participants or dishonest raters. Finally, we argue that the simple local trust formula used by EigenTrust, though capturing the quantitative difference between the number of satisfactory transactions and the number of unsatisfactory transactions between a pair of participants, fails to reflect the total number of transactions that the pair of participants have had, enabling malicious participants to manipulate the EigenTrust system at low cost.

In this paper, we first analyze when EigenTrust is effective and when it may fail badly. Based on the analysis, we present the design of EigenTrust^{++}, the attack resilient reputation trust management for a collaboration network. EigenTrust^{++} enhances the attack resilience of EigenTrust from three core aspects. First, EigenTrust^{++} makes a clean separation of transactional experience based reputation from feedback referral based reputation to increase the attack resilience in the presence of dishonest feedbacks. Second, EigenTrust^{++} removes the uniform trust propagation model in the reputation computation and instead introduce a linear threshold based probabilistic trust propagating model, with personalized similarity as the edge weights. This allows EigenTrust^{++} to activate the trust propagation from a participant to only those of its neighbor participants which have similar transactional and feedback behavior. Thus, EigenTrust^{++} makes it much harder for malicious participants to gain trust from (be activated by) good peers, and enables good participants to share their experiences and feedbacks with their circles of the friends that have similar transactional and feedback behaviors. By making the trust propagation model more attack resilient in the presence of dishonest feedbacks, sparse networks and large number of malicious peers, EigenTrust^{++} can significantly cut down the propagation paths from good peers to malicious peers. Third but not the least, EigenTrust^{++} also strengthens the attack resilience of the local trust value by incorporating the total number of transactions performed between a pair of participants in addition to the difference between satisfied and unsatisfied number of transactions. We conduct extensive experimental evaluation of EigenTrust^{++} in terms of its effectiveness against four representative threat models with varying percentages of dishonest feedbacks.

The rest of this paper is organized as follows. Section II gives an overview of the core components of EigenTrust, identifies its inherent vulnerabilities and analyzes the root causes of such vulnerabilities. We describe EigenTrust^{++} in Section III, report our experimental evaluation results in Section IV, discuss related work and conclude in Section V.

II. EIGENTRUST AND ITS INHERENT VULNERABILITIES

In this section we first give a brief review of EigenTrust and the four popular threat models introduced by EigenTrust to make the paper self-contained. Then we analyze the vulnerabilities inherent in EigenTrust from local trust value computation, global trust aggregation and the problem of using uniform probability distribution in its eigenvector based trust propagation model and outline the design of EigenTrust^{++}.

A. EigenTrust Overview

EigenTrust is a simple and intelligible reputation management system with three core components, local trust value, aggregation of local trust value into global reputation score, and hop based trust propagation that converges to the principal left eigenvector through uniform distribution of propagation probability.

In most reputation systems, participants (peers) interact with one another to provide services, be it query-answering or tweeting or blogging. Thus, we refer to the participant who requests a service as client or consumer and the participant who offers the service as the server or producer. Upon the completion of a transaction between a pair of participants, the consumer peer will rate the producer peer in terms of its transaction quality, denoted by $tr(i, j)$, initialized to zero. In EigenTrust, a peer i may rate the transaction it has with peer j as positive by $tr(i, j)$ = 1 or negative by $tr(i, j)$ = -1.

The transaction based local trust from peer i to peer j, denoted by s_{ij}, is defined in EigenTrust as the difference between satisfactory transactions, $sat(i, j)$, and unsatisfactory transactions, $unsat(i, j)$, between peer i and peer j. Namely sij = $sat(i, j)$ - $unsat(i, j)$. To prevent malicious peers to assign arbitrarily high local trust values to other malicious peers, and arbitrarily low local trust value to good peers, a normalized local trust value that peer i has over peer j, denoted by c_{ij}, is used in EigenTrust, which normalizes s_{ij} by the maximum satisfactory score from all peers who have had the direct transactional experiences with peer i as follows:

$$c_{ij} = \frac{max(s_{ij},0)}{\sum_k \max(s_{ik},0)}, \ \ if \sum_k \max(s_{ik}, 0) \neq 0$$

$$c_{ij} = p_j \qquad\qquad Otherwise$$

EigenTrust uses the normalized local trust value c_{ij} as the initial trust for each pair of participants who have direct transactions or interactions. When a peer i does not have any transaction with anyone in the system, we have s_{ij} =0 for any

j and set $c_{ij} = p_j$, where $p_j = 1/|P|$ if $j \in P$ otherwise $p_j = 0$. P is a set of pre-trusted peers and is used as the central authority of the system. Otherwise, s_{ij} is normalized by using the sum of the local trust values of all neighbors of peer i as the denominator.

Let n denote the total number of participants in the system, we can define C as the matrix $[c_{ij}]$ with n rows and n columns. Let $\vec{t}^{k+1}$ denote the global reputation trust vector of size n to be computed at k + 1 round of iterations. We define $\vec{t}^{k+1} = (1-\alpha)C^T\vec{t}^{k} + \alpha\vec{p}$, where α is the probability of a peer knows none and thus relies on the pre-trusted peers, and denotes the initial trust vector with only pre-trusted peers having the reputation value of $1/|P|$. Alternatively, a peer i's reputation at (k+1)th iteration can be defined by the sum of the local trust values that other peers have given to i, weighted by their trust score at the kth iteration, namely $t_i^{(k+1)} = (1-\alpha)(c_{1i}t_1^{(k)} + \cdots + c_{ni}t_n^{(k)}) + \alpha p_i$.

B. Threat Models and Trust based Service Selection Methods

One way to evaluate the reputation trust models is to measure its resilience to different attack strategies of malicious participants. The following six threat models are documented in EigenTrust [7].

Threat Model A. Independently Malicious. Malicious peers are independent and not initially aware of other malicious peers and simply upload inauthentic files or provide bad services.

Threat Model B. Chain of Malicious Collectives. Malicious peers know each other upfront and deterministically give a high local trust value, say 1, to another malicious peer, resembles a malicious chain of mutual high local trust values. Malicious peers always provide an inauthentic file or bad service when selected as download source or service provider.

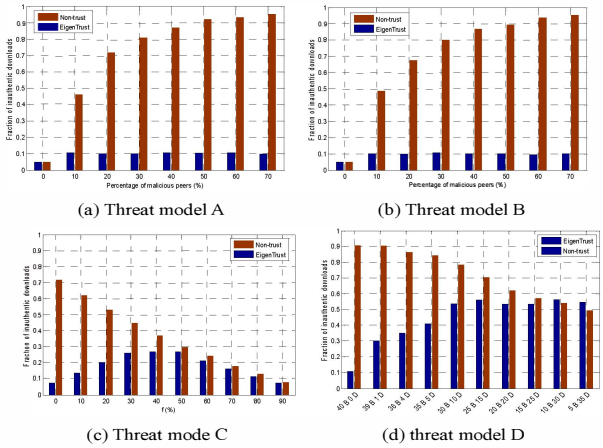
Threat Model C. Malicious Collectives with Camouflage. Malicious peers try to get some high local trust values from good peers by providing authentic files in $f\%$ of all cases when selected as download sources or service providers.

Threat Model D. Malicious Spies. Malicious peers are strategically organized into two groups. One group of malicious peers (type D) try to act as normal peers in the network and try to increase their global reputation by only providing authentic files or good services, and uses the reputation they gain to boost the trust values of another group of malicious peers (type B) who only provide inauthentic files or bad services when selected as download sources or service providers.

Threat Model E. Sybil Attack. A malicious peer initiates thousands of peers in the network. Each time one peer is selected for download or as a service provider, it sends an inauthentic file or provides a bad service and then disconnect and replaced with a new identity.

Threat Model F. Virus-Disseminators. A malicious peer sends one inauthentic virus infected file every 100th request. All other times, it sends an authentic file. This is a variant of Threat Model C.

Figure 1 shows the performance of EigenTrust in Threat Models A, B, C and D.



(a) Threat model A (b) Threat model B

(c) Threat mode C (d) threat model D

Fig. 1. EigenTrust Performance in Threat Models A, B, C and D

Deterministic v.s. Probabilistic Service Selection. Given a query service request (say download of a particular piece of music) and a subset R of peers with matching results, there are two reputation based service selection criterions to choose a peer with high reputation to serve. The deterministic algorithm chooses the peer that has the highest reputation among those responding peers as the download source. The probabilistic algorithm chooses a peer i as download source according to the trust based probability $t_i / \sum_{j=0}^{R} t_j$, ensuring that a peer with higher reputation score will have higher probability to be selected. To do this, a decimal in interval [0,1] needs to be generated randomly, then, this decimal is utilized to compare with the trust base probability of response peer i, if the probability is bigger than this decimal, peer i would be selected as service provider, if not, another response peer is chosen and compared again. In EigenTrust, the probabilistic algorithm is used as the reputation-based service selection method, augmented by the following refinement: with the default probability of 10%, EigenTrust selects one of those peers whose reputation values are zero as the download source [7]. The refined probability-based selection gives new participants some chance to build up reputation, and at the same time prevents the system from overloading peers with high reputation scores.

C. Vulnerabilities Inherent in EigenTrust

We examine vulnerabilities inherent in the EigenTrust model: Local Trust Rating, Feedback Credibility, and Uniform Trust Propagation Distribution.

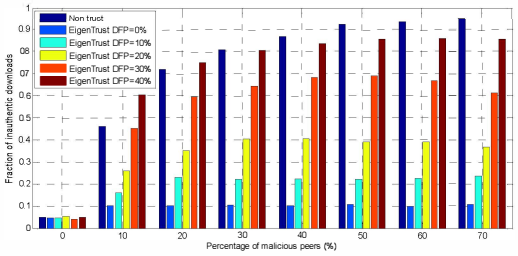
Local Trust Rating. The local trust value in EigenTrust is defined as $s_{ij} = sat(i,j) - unsat(i,j)$, which is the difference between the number of satisfactory transactions peer i has had with peer j and the number of unsatisfactory transactions. Unfortunately, this formula suffers from a number of vulnerabilities. Consider the following three scenarios: (a) peer i has 10000 satisfactory and 9980 unsatisfactory transactions with j_1; (b) peer i has 100 satisfactory and 80 unsatisfactory transactions with peer j_2; (c) peer i has 20 satisfactory and 0 unsatisfactory transactions with j_3. Clearly by the formula above, we have that $s_{ij_1} = s_{ij_2} = s_{ij_3} = 20$, namely, their

local trust values are equal. However, it is obvious that the behavior of peer j_3 is much better than that of j_2, and the behavior of j_2 is better than that of j_1 to some extent. A cause for such vulnerabilities is the lack of consideration on how the total number of transactions may enhance the attack resilience of EigenTrust local trust computation.

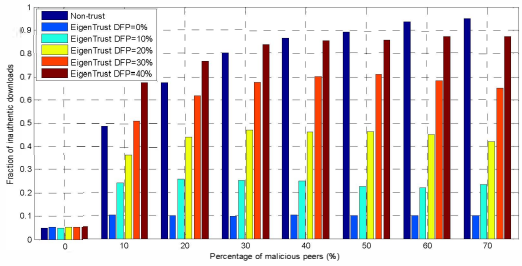
Feedback Credibility. Figure1 shows that EigenTrust is effective in the presence of varying percentage of malicious peers up to 70% in Threat models A and B where either malicious peers are independently malicious or forms a malicious chain of high feedback ratings but only somewhat effective when malicious peers are up to 20-40% in Threat models C and D. However, these measurements are conducted under the assumption that good peers are good for both transactions and referrals, which is unrealistic in practice. For instance, a peer may make false statements about another peer's service due to jealousy or other types of malicious motives. Consequently, a trustworthy peer may end up getting a large number of false statements and its reputation is evaluated incorrectly even though it provides satisfactory service in every transaction [18]. We argue that the effectiveness of a reputation trust should be resilient in the presence of varying dishonest feedback percentage (DFP) by good peer, namely with certain probability, peers performing good transactions may provide dishonest feedbacks.

Figure 2 shows the measured EigenTrust efficacy in terms of inauthentic downloads under different DFP settings from zero to 40% for Threat Models A and B with the percentage of malicious peers up to 70%. Both Figure 2(a) and Figure 2(b) are the measurement results of EigenTrust with the probabilistic selection method using the same simulation setup as [7]. When the DFP is set to 0%, all good peers give honest feedback ratings, we measure EigenTrust and obtain the almost same results as reported in [7] for both non-trust case and EigenTrust DFP=0% (shown in the first two blue bars under each fraction of malicious peers). However, when the DFP changes from 10% to 40%, the fraction of inauthentic downloads goes up quickly and proportionally. More interestingly, the fraction of inauthentic downloads is higher than the non-trust case for EigenTrust when the DFP reaches 40% and the malicious peers are from 10% to 30% in Threat models A and B. This set of experiments shows some serious vulnerabilities of EigenTrust in the presence of dishonest feedbacks for good peers. With the increase of malicious peers, for non-trust case in which a consumer randomly select a service providers without referring to reputation measurement, the probability that malicious peers are selected as download sources would become large proportionally. Thus, the fraction of inauthentic downloads should rise as well. This is the reason why EigenTrust only slightly improves over the non-trust case when the fraction of malicious peers is 40% or higher 70% but worse when the malicious peers are from 10% to 30%.

Utilizing Circle of Friends. EigenTrust utilizes power iteration to compute the reputation value for each peer based on the normalized local trust matrix C. A large number of iteration rounds can guarantee the computation stability since



(a) Threat Model A based on Probabilistic Algorithm



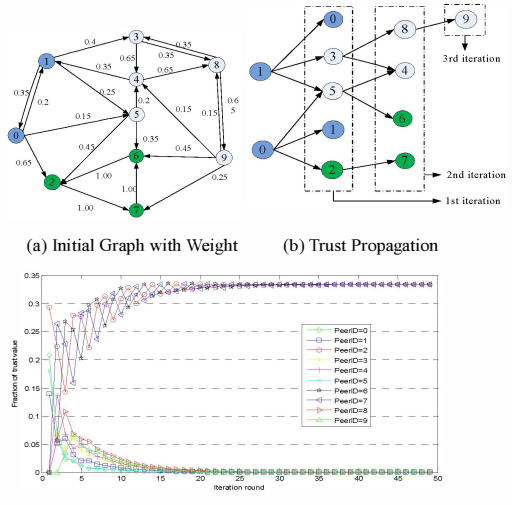
(b) Threat Model B based on Probabilistic Algorithm

Fig. 2. The Fraction of Inauthentic Downloads in EigenTrust

it converges to the left principle eigenvector of local trust matrix C. Although power iteration is effective for propagation based trust computation, one key challenge is to determine how many iterations should be implemented.

Figure 3(a) shows an example network of 10 nodes, where nodes 0 and 1 are pre-trusted peers with initial reputation values $1/|P|$. By the EigenTrust formula $\overrightarrow{t}^{(k+1)} = (1-\alpha)C^T\overrightarrow{t}^{(k)} + \alpha\overrightarrow{p}$, we calculate reputation value for each node iteratively up to $k = n = 100$ rounds. Figure 3(b) illustrates the first three rounds of trust propagation process. In each iteration round, a node can compute its reputation based on the reputation values of those nodes from which it can be reached in one hop. At the same time, this node can propagate its reputation value to its direct neighbors. Consider this example network, at first, only nodes 0 and 1 have non-zero reputation value of 1/2. After finishing the 1st iteration, nodes 2 and 3 receive trust values from nodes 0 and 1 respectively, and node 5 receives trust values from the both nodes 0 and 1. Similarly, nodes 4, 6, 7 and 8 receive trust values after finishing the 2nd iteration. Node 9 would receive reputation propagation after the 3rd iteration. If the iteration round k is large enough, the computation will converge to the left principle eigenvector of matrix C. Figure 3(c) shows the simulation results with the parameter increment ε set to 0.0001.

Note that nodes 2, 6 and 7 form a chain with high weight value on each edge. This scenario corresponds to the threat mode C of malicious collectives with Camouflage: on one hand, they provide authentic files and gain positive local trust values, such as node 2 gets positive rating from pre-trusted node 0 and node 5, and on the other hand, they always give exaggerated rating to their malicious neighbors and negative rating to good neighbor peers. Fig. 3(c) shows that the reputation values of nodes 2, 6 and 7 are increased



(a) Initial Graph with Weight (b) Trust Propagation

(c) Reputation with Increasing Iteration Rounds

Fig. 3. EigenTrust Reputation Propagation Model

visibly, while the other peers' reputation values decline as we increase the number of iterations. This clearly violates the goal of reputation trust management, which is to promote the reputation of good peers and reduce the reputation of malicious peers.

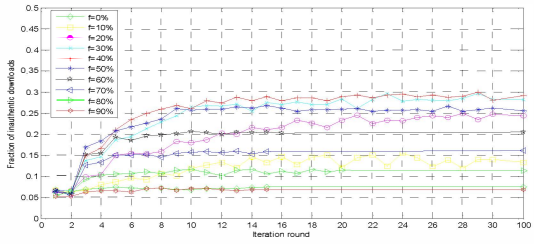


Fig. 4. Fraction of Inauthentic Downloads with Iteration Rounds

Figure 4 shows the fractions of inauthentic download of Threat Model C for each iteration round. Malicious peers try to get some high local trust values from good peers by providing authentic files in $f\%$ of all cases when selected as download sources or service providers. The simulation result is consistent with the one reported in [7], shown in Figure 1 (c). As the iteration round increases, the fraction of inauthentic downloads goes up. This indicates that when lots of dishonest ratings exist, the power iteration not only fails to promote the trust level of good participants but also starts to help in a reverse direction, boosting the reputation of the malicious peers such as nodes 2, 6 and 7 in Figure 3(c).

Figure 5 shows how the reputation changes as the iteration rounds increases for selected good (G), pre-trusted (P) and malicious (M) peers under Threat model C, in which the percentage of malicious peers is set to 27%. We observe that when the probability f that malicious peer provides authentic services are increased to 30% or higher to 90%, all peers' reputation values change significantly except the pre-trusted peers. With higher f, malicious peers can obtain more positive rating thus gain larger reputation, and they in turn give positive ratings to the chain of malicious partners, which builds a high local trust value between a pair of malicious peers in the chain. With certain number of iteration rounds, the malicious peers can obtain more reputation via the reputation propagation. At the same time the reputation values of good peers drop as more malicious peers gain high reputation. We also observe that when the probability f is lower than 30%, the change in reputation value is slow and even when f is from 30% 0r 40%, the increasing rate of malicious peers' reputation values are still smaller than that of good peers. The reputation values of malicious peers climb evidently and are larger than that of good peers, when f is 90%. Several factors (e.g., the network size, its topological structure, percentage of malicious nodes, threat model) determine the exact threshold of f and the exact iteration round when the gap of reputation between malicious and good peers also becomes larger. Therefore, good tuning of the reputation propagation model in terms of how many iterations are sufficient and how to propagate trust among peers who have no prior transactions is critical to the efficiency and effectiveness of a reputation trust model.

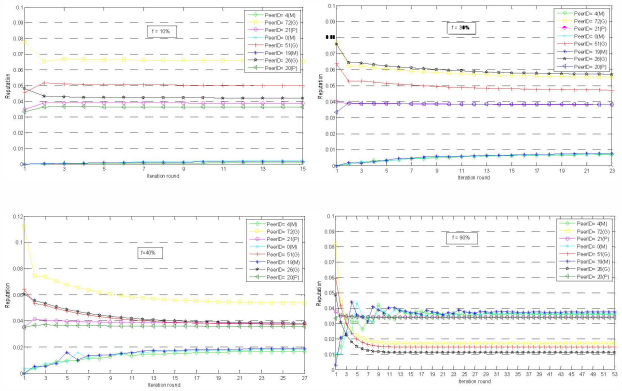


Fig. 5. The Reputation of Good, Malicious and Pre-trusted Peers as the Iteration Rounds Increase with Varying f

We have analyzed and identified three types of vulnerabilities in EigenTrust. Before we introduce our solution for attack resilient trust management, EigenTrust^{++}, in the next section, we would like to note that the rating density may also have significant influence on the effectiveness of trust propagation. In EigenTrust simulator, both pre-trusted peers and malicious peers have at least 10 initial neighbors, and good peers have at least 2 initial neighbors, 7 hops are set as search range. However, the peer to peer network is only used for answering queries. The trust network among all participant peers is established based on direct transaction ratings and the corresponding normalized local trust values. Thus, the trust propagation is conducted over the trust network. When this trust network is sparse, namely most peers have only transactions with a very small number of other peers, There will be a good portion of peers with zero as their reputation values if the iteration round is set to only a small number of hops. However, when the density of the trust network is

relatively high, if the iteration round is set to a larger value, then, malicious participants may gain more advantages than intended.

III. EIGENTRUST^{++}: ATTACK RESILIENT TRUST METRICS

In this section, we present EigenTrust^{++}, an attack resilient trust management model. It replaces three core components of EigenTrust to counter the vulnerabilities of EigenTrust [7]. First, we define a more sophisticated transaction rating formula to compute the local trust by introducing the total amount of transactions between a pair of participants as the denominator. Second, we replace the initial global trust formula that aggregates local trust values by a feedback credibility based trust metric, which utilizes personalized rating similarity to aggregate local trusts. Third, we replace the uniform distribution based trust propagation model by a differential probabilistic trust propagation model. We use a linear threshold to control how the trust propagation is conducted among peers to allow a participating peer to distribute more trust to its neighbor peers whom he or she has higher confidence based on personalized rating similarity. Finally, the computation complexity is discussed.

A. Attack Resilient Local Trust Computation

In EigenTrust, the local trust value that a peer i places on another peer j, denoted by s_{ij}, is defined by the difference between the number of satisfactory transactions and the number of unsatisfactory transactions. However, this definition suffers from some serious vulnerability. For example, it treats the 10 satisfactory transactions the same as 100 satisfactory and 90 unsatisfactory transactions combined. Similarly, it treats 1000 satisfactory and 998 unsatisfactory transactions the same as 4 satisfactory and 2 unsatisfactory transactions, ignoring the fact that the former should be less trust-worthy than the latter due to its excessive number of unsatisfactory transactions, even though the absolute difference is the same. Therefore, we revise the transaction based local trust formula by introducing the total number of transactions as the denominator:

$$s_{ij} = \begin{cases} \frac{sat(i,j)}{sat(i,j)+unsat(i,j)} & if \;\; sat(i,j) + unsat(i,j) > 0 \\ 0 & otherwise \end{cases} \tag{1}$$

To facilitate the comparison among different local trust values, we also normalize the local trust values as follows:

$$c_{ij} = \begin{cases} \frac{max(s_{ij},0)}{\sum_m max(s_{im},0)} & if \;\; \sum_m max(s_{im},0) \neq 0 \\ p_j & otherwise \end{cases} \tag{2}$$

Here p_j represents the set P of pre-trusted peers, and $p_j = 1/|P|$ if $j \in P$, and $p_j = 0$ otherwise.

B. Aggregating Local Trust Values Using Feedback Credibility

On one hand we have shown in Figure 2 that EigenTrust is no long effective in the presence of dishonest feedbacks. On the other hand, dishonest feedbacks exist in many social media and eCommerce reputation systems, e.g. Amazon, eBay. It is quite possible for malicious peers to perform good transactions strategically to gain high reputations by collecting positive feedback ratings but provide dishonest feedbacks to other good peers. When malicious peers collude in such manner, EigenTrust becomes vulnerable against strategic manipulation (such as Threat Models C, D). It is worth noting that the high percentage of inauthentic downloads shown in Figure 2 is mainly induced by the positive ratings collected by malicious peers from other good or malicious peers. Therefore, how to differentiate the positive ratings generated by good peers from those generated by malicious peers is crucial. This motivates us to introduce the feedback similarity metric to define and measure the feedback credibility of a peer. This is motivated by the observation that two good peers may give very similar feedback ratings to the same common set of peers with which they have had interactions or transactions in the past. Two malicious peers, on the other hand, may give very similar feedback ratings to the same common set of good peers with which they have had transactions. On the contrary, a good peer and a malicious peer will give very different feedback ratings to the same set of peers they have interacted with. We define feedback credibility by utilizing such feedback similarity metric as a weight to the local trust value. A peer has high global reputation only if he has received both high feedback credibility and high local trust from other peers. For example, a local trust value that a malicious peer i places on a good peer j should be weighted by the feedback credibility of peer i.

Given that each peer can provide feedback to other peers with whom he or she has had direct interaction or transaction, thus we can assign one feedback vector of size N for each peer, and N is the total number of participants in the reputation system. To compute the similarity between two feedback vectors, we can use one of the conventional vector similarity metrics, e.g., cosine coefficient, Jaccard's coefficient, distance-based and standard deviation-based metrics. In the first version of EigenTrust^{++}, we compute the feedback similarity by utilizing the standard deviation-based method, which shows how much variation or "dispersion" exists from the expected value or the evaluated individuals in our context. The larger the standard deviation, the smaller the similarity, thus the feedback similarity between two peers u and v is defined as follows:

$$\begin{aligned} sim(u,v) &= 1 - \sqrt{\frac{\sum_{w \in comn(u,v)} (tr(u,w) - tr(v,w))^2}{|comn(u,v)|}} \\ tr(u,w) &= \sum_{i=1}^{|R(u,w)|} tr_i(u,w)/|R(u,w)| \\ tr(v,w) &= \sum_{i=1}^{|R(v,w)|} tr_i(v,w)/|R(v,w)| \end{aligned} \tag{3}$$

$comn(u,v)$ denotes the subset of the common peers that have had interaction with both peer u and peer v and $R(u,w)$ is the number of transactions between peer u and peer w. The feedback similarity of any two peers depends on the historical local trust ratings that are received by those common peers. Also when two peers have had many transactions, there

can be many different ratings. We calculate the local trust for each finished transaction and average them, instead of one-time computation, and normalize it by the total number of satisfactory and unsatisfactory transactions that have been done between the two transacting peers. Based on the feedback similarity, we define the feedback credibility f_{ij} as follows:

$$f_{ij} = \frac{sim(i,j)}{\sum_{m=1}^{R(i)} sim(i,m)} \quad (4)$$

Here $R(i)$ is a set of the peers that have had transactions with peer i. Thus, we define the feedback credibility that peer i place on peer j, denoted by fc_{ij}, as follows:

$$fc_{ij} = f_{ij} \cdot c_{ij} \quad (5)$$

Again to facilitate the comparison of different feedback credibilities, we normalize fc_{ij} as follows:

$$l_{ij} = \begin{cases} \frac{max(fc_{ij},0)}{\sum_{m=1}^{R(i)} max(fc_{im},0)} & if \sum_{m=1}^{R(i)} max(fc_{im},0) \neq 0 \\ 0 & otherwise \end{cases} \quad (6)$$

Below we define the initial trust that peer i place on peer j, denoted by t_{ij}, by the local trust values received by peer j from other peers, say k, weighted by the feedback credibility peer i places on peer k :

$$t_{ij} = \sum_k l_{ik} c_{kj} \quad (7)$$

With iterative computation of global reputation, we can compute the (k+1)th iteration by utilizing the trust values computed at kth round as follows:

$$t_i^{(k+1)} = (1-\alpha)(l_{1i} \cdot t_1^{(k)} + l_{2i} \cdot t_2^{(k)} + \cdots + l_{ni} \cdot t_n^{(k)}) + \alpha p_i \quad (8)$$

This formula can effectively restrain the positive ratings from malicious peers via similarity-based feedback credibility.

C. Probabilistic Trust Propagation through Circle of Friends

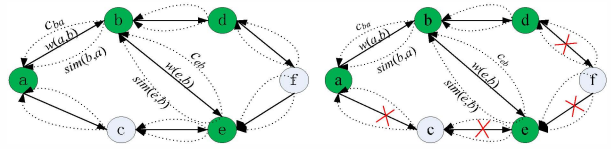
It is widely recognized in most social media systems that participating peers often do not have the same response with respect to the same stimuli or sensing results. The uniform distribution based trust propagation in EigenTrust is unrealistic and also vulnerable because it may lead to boost the reputation of malicious peers as shown in Figure 2 when dishonest feedbacks exist. We argue that an ideal trust mechanism should provide the controlled trust propagation via the circle of friends, which can encourage trust propagation to good peers and cut off the trust propagation to malicious peers. This requires a method to determine which next hop neighbors should participate in trust propagation and which should not. However, it is hard to provide a clean separation of good peers from malicious peers in the presence of different types of malicious behavior and attacks, such as cheating, collusion, camouflage and spy, etc. In this section, we describe how we replace the uniform distribution based trust propagation to a probabilistic linear threshold based model that can dynamically control the trust propagation to the circle of friends based on both the level of local trust and the feedback similarity.

Combining local trust and feedback similarity. Concretely, in EigenTrust^{++}, instead of using the local trust as the edge weight for trust propagation between two peers, we propose to use a weighted combination of local trust and feedback similarity, namely $w(i,j) = (1-\beta) \times c_{ji} + \beta \times sim(j,i)$. The proportion factor β is used to balance the two factors, with β probability using feedback similarity and 1-β probability using local trust between two peers that have direct transactions. Note that the normalized local trust value c_{ij} reflects the direct trust peer i has in peer j. In fact, c_{ij} also indicates the confidence that peer i has in peer j. However, malicious peers may strategically perform good transactions to gain high local trust values under Threat models C and D, thus propagating trust based solely on local trust value may not be attack resilient. The intuition is to identify who are malicious and who are good, then cut off the positive ratings to malicious ones and simultaneously prevent the malicious ratings to good peers. For good peers, an effective way is to find those peers that have a similar behavior and propagate positive trust ratings to only those neighbor peers they have high feedback similarity and high local trust value. Therefore we introduce the feedback similarity as another parameter to help control and balance the trust propagation to the right peers. On the other hand, only relying on the feedback similarity without taking into account the local trust values can also lead to unwanted vulnerability. However, comparing with local trust we should focus more attention on the feedback similarity and set β as 0.85.

Threshold-based Probabilistic Trust Propagation. In EigenTrust^{++}, we requires that the edge weight must be subjected to $\sum_{u \in V} w(u,v) \leq 1$ where V denotes the set of participating peers, and $w(u,v)$ denotes the weight on the edge of (u,v). When given an initial active seed set $S \subseteq V$, the other vertices that are not in this set would be regarded as inactive vertices. The trust propagation would be performed as follows: first, each vertex uniformly chooses a threshold at random from interval [0,1]. This represents the weighted fraction of v's neighbors that must become active in order for v to become active. Then, the propagation process unfolds deterministically in discrete steps. In step i, all its active neighbors is at least θ_v would be activated: $\sum_{u \in AN(v)} w(u,v) \geq \theta_v$ where $AN(v)$ denotes the set of active neighbors of vertex v. The thresholds θ_v intuitively represent the different latent tendencies of vertices to adopt the trust when their neighbors do. The fact that they are randomly selected is intended to model the lack of knowledge of their true values. The choice of the threshold can be set either by 1/2 or randomly selected from [0,1]. One way to further increase the attack resilience is to use statistical methods to guide the setting of this threshold.

Consider the example displayed in Figure 6(a). Each edge has two weights: local trust and feedback similarity. By combining these two metrics, we can utilize a system supplied threshold to control when and how to propagate the current trust and to whom such trust should be propagated. Concretely, we set node a as the initial active node, and start the trust propagation step by step. When termination, assuming that nodes b, d and e become active, and nodes c and f are

inactive, because they have a small local trust rating or a low similarity with the active nodes. In EigenTrust^{++}, only the active nodes are regarded as the circle of friends to propagate for node a as in Figure 6(b), which also means to cut off the trust propagation path to inactive nodes. There are two situations for consideration: first, if node a is good peer, then active peers may be good peers as well, so propagating trust to them is a rational. The inactive nodes may be malicious peers, discarding the propagation path to them is making good sense. Second, if node a is a malicious peer, the active nodes may be malicious peers, the malicious form a collective, they propagate trust to their partners. Since they cannot obtain reputation from the good peers, the trust propagation is weak among these malicious peers. The inactive may be good peers, and by cutting off the path from malicious peers helps preventing the dishonest ratings to be propagated to the good peers. This illustrates why EigenTrust^{++} threshold based probabilistic trust propagation method works effectively.



(a) Initial Graph with Local Trust and Similarity (b) Controlled Trust Propagation

Fig. 6. EigenTrust^{++} Probabilistic Threshold Controlled Trust Propagation via Circle of Friends

In EigenTrust^{++}, we aggregate the local trust ratings by asking the acquaintances about other peers. For example, peer i should ask its friends, its friends' friends ($\overrightarrow{t}_i = (C^T)^2\overrightarrow{c_i}$) and then propagate trust among this trusted community. The relationship of trust propagation over the transactional/collaboration network can be expressed via the matrix C, and AC presented the activation matrix of eight. Now we transform the matrix C as AC below:

$$AC = \begin{bmatrix} a_1^1 \cdot c_{11}, & a_2^1 \cdot c_{12}..., & a_n^1 \cdot c_{13} \\ a_1^2 \cdot c_{21}, & a_2^2 \cdot c_{22}, ..., & a_n^2 \cdot c_{2n} \\ \vdots & \ddots & \vdots \\ a_1^n \cdot c_{n1}, & a_2^n \cdot c_{n2}, ..., & a_n^n \cdot c_{nn} \end{bmatrix}$$

$$a_j^i = \begin{cases} 1, & if\ i\ make\ j\quad active \\ 0, & otherwise \end{cases}$$

To guarantee all the values between 0 and 1, we also normalize matrix AC as:

$$ac_{ij} = \begin{cases} \frac{a_j^i \cdot c_{ij}}{\sum_m a_m^i \cdot c_{im}} & if\ \sum_m a_m^i \cdot c_{im} \neq 0 \\ 0 & otherwise \end{cases} \quad (9)$$

Therefore, we propose the following matrix expression to aggregate local trust:

$$\overrightarrow{t_i} = (AC^T)^{(n)} \cdot \overrightarrow{ac_i} \quad (10)$$

In the local trust vector $\overrightarrow{t_i}$, it has its own experience in peer k not by asking its friends. In this instance, we should take both direct trust and indirect (friends') trust into account, and assign a proportion parameter to adjust them.

$$t_{ik}^{(k+1)} = (1-\gamma)(ac_{i1}^{(k)} \cdot ac_{1k} + ac_{i2}^{(k)} \cdot ac_{2k} + \cdots + ac_{in}^{(k)} ac_{nk}) + \gamma \cdot ac_{ik} \quad (11)$$

Here the proportion factor γ is utilized to balance the direct and indirect trust.

Algorithm Complexity Analysis. In EigenTrust^{++}, the main overhead is the computation of global reputation using the threshold-based probabilistic trust propagation model. Thus the time complexity relies on the number of iterations for computing global reputation and the size n of the network (total number of participants). In EigenTrust, for each peer, its reputation is aggregated by simply asking other n-1 peers and thus for n peers, the computation complexity is $O(n^2)$. In EigenTrust^{++}, it needs $O(1)$ time to check whether a peer become active or not. A peer aggregates its reputation by asking only its active neighbors. The loop will continue until no peer can be activated by its active neighbors. Thus it will take no more than n runs to terminate. In EigenTrust^{++}, each peer has an active list, so the complexity is also $O(n^2)$.

IV. EXPERIMENTS AND RESULTS

In this section, we evaluate the efficacy of EigenTrust^{++} by comparing it with EigenTrust [7] in terms of efficiency, effectiveness and attack resilience. To make a fair comparison with EigenTrust, we build a simulator on top of the simulation platform TM/RM simulator [12] and incorporate all four attack models used in EigenTrust [7] into the TM/RM simulator to evaluate and compare the performance of EigenTrust^{++} with EigenTrust from different perspectives. Figure 1 shows the experimental results of EigenTrust, which reproduce the almost same experimental observations reported in the original EigenTrust paper [7].

A. Parameter Configuration

EigenTrust^{++} can be seen as an attack resilient enhancement of Eigentrust. To make a fair comparison with original EigenTrust, we set the similar configuration parameters as those chosen in EigenTrust experiments reported in [7], ranging from network structure, file distribution to peer behavior. Table I shows the list of the parameters.

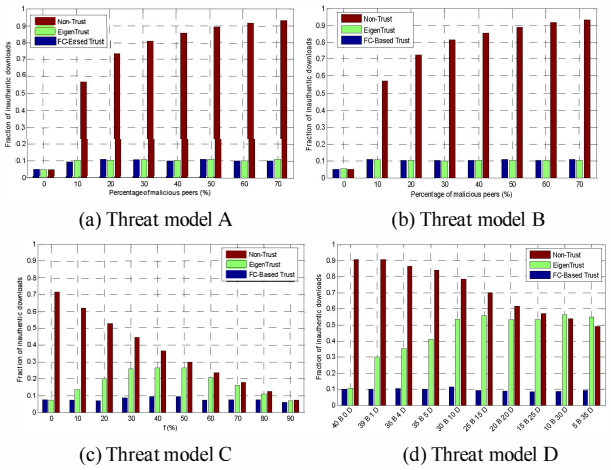
The query service network is setup in a similar fashion as [7]. Both malicious and pre-trusted peers have 10 initial neighbors, and good peers have 2 initial neighbors. Originally only pre-trusted peers have positive reputation. When a query is issued by a peer, it is propagated by the scoped broadcast mechanism with hop-count horizon over the entire network in the usual Gnutella way. Peers that receive the query forward it to the next hop peers and also check whether they have the requested file or not, if have, respond it. Similar to [7], we set 7 hops as the default response range. Furthermore, the number of distinct files is assigned to each peer according to Zipf distribution, and popular files have more copies in the system. On the other hand, the number of queries issued for different files are also based on Zipf distribution.

TABLE I
SIMULATION CONFIGURATION

Network Structure	number of good peers	60-100
	number of pre-trusted peers	3
	number of initial neighbors of good peers	2
	number of initial neighbors of malicious peers	10
	number of initial neighbors of pre-trusted peers	10
	number of hops for query process	7
File Distribution	file distribution at good peers	Zipf distribution over 20 distinct files
	number of distinct files at good peer	Uniform random distribution
	top % of queries for most popular files malicious peers respond to	20%
	top % queries for most popular files pre-trusted peers respond to	5%
	% file categories owned by good peers in threat model A, B and C	15%
	% file categories owned by good peers in threat model D	10%
	% file categories owned by malicious peers in threat model A, B, D	100%
	% of file categories owned by malicious peers in threat model C	30%
Peer Behavior	% of download requests in which good peer returns inauthentic file	5%
	Downloads source selection algorithm	probabilistic algorithm
	Probability that peer with global trust value 0 is selected	range [1%-10%]

B. *Effectiveness of Feedback Credibility*

In this section, we verify the effectiveness of EigenTrust^{++} by comparing it with EigenTrust, focusing on understanding the effect of feedback credibility on attack resilience of EigenTrust^{++}. All our experiments are running simulations under the four threat models A, B and C and D. We compare EigenTrust^{++} feedback credibility (FC) with EigenTrust and non-Trust. The experimental results are depicted in Figure 7. It is clear that the feedback credibility (FC)-based trust metric can constraint the trust propagation to malicious peers from good peers because of their limited similarity. Thus it outperforms EigenTrust under all the four threat models.



(a) Threat model A (b) Threat model B

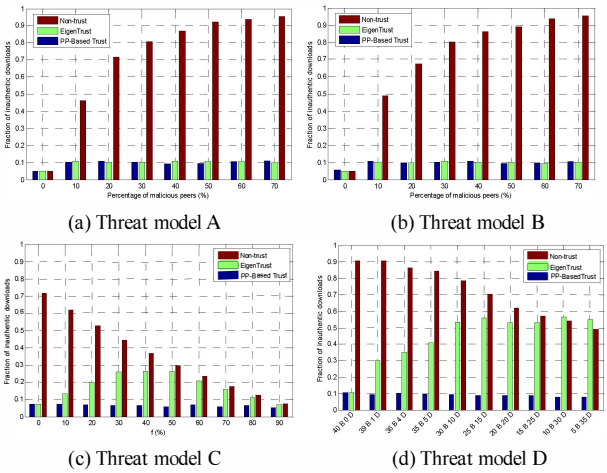
(c) Threat model C (d) Threat model D

Fig. 7. Fraction of Inauthentic Downloads in the Four Threat Models

In Threat models A and B, EigenTrust^{++} FC-based trust management offers the same level of efficiency and effectiveness compared to the results in EienTrust. This is because the malicious cannot provide authentic uploading services, thus they cannot gain positive ratings from good peers, even they can obtain positive ratings from other malicious partners in threat model B, their reputation values are still zero since their partners' reputation values are zero. However, in Threat model C, the camouflage peers can get the positive ratings from good peers since they can provide authentic content services. Then they give high local trust values to their partners based on chain. Figure 7(c) shows that compared with EigenTrust, the FC-based trust metric in EigenTrust^{++} can significantly outperform EigenTrust (recall Figure 1). This demonstrates that the FC-based trust mechanism can prevent the trust from propagating to camouflage peers. In Threat model D, the type D peer can provide authentic content services for other peers and obtain positive ratings, by which they gain reputation, and then give high local trust values to all the type B peers. Figure 7(d) shows that the FC-powered EigenTrust^{++} mechanism obviously outperforms EigenTrust and non-trust, which elaborates that our FC-powered EigenTrust mechanism confines the trust propagation to malicious peers effectively.

C. *Effectiveness of Circle of Friends*

We evaluate the effectiveness of EigenTrust^{++} via the comparison with EigenTrust and non-trust case. Figure 8 shows the experimental results for the four threat models A, B, C and D by comparing EigenTrust^{++} with EigenTrust and non-Trust. It is clear that the probabilistic propagation (PP)-based trust metric can effectively constraint the trust propagation to malicious peers from good peers. Thus EigenTrust^{++} significantly outperforms EigenTrust in the threat models C and D where malicious collectives collude while EigenTrust^{++} performs equally well as EigenTrust in the threat models A and B.



(a) Threat model A (b) Threat model B

(c) Threat model C (d) Threat model D

Fig. 8. Fraction of Inauthentic Downloads in the Four Threat Models

There are a number of reasons why EigenTrust^{++} offers remarkable efficiency and effectiveness and EigenTrust original does poorly (recall Figure 1) in the threat models C and D where the camouflage and type D peers offer authentic content services and thus gain positive reputations. First, colluding malicious peers give high local trust values to one another. Second, EigenTrust^{++} replaced the simple trust aggregation and trust propagation mechanisms with similarity based feedback credibility and threshold-based probabilistic propagation to prevent the trust of good peers being propagated to camouflage and type B peers.

V. RELATED WORK AND CONCLUSION

Reputation systems have been used in practice, such as eBay, the popular online auction system. The research on Reputation systems have been active in several fields, from distributed computing, peer to peer network computing, cloud computing, social network and mobile networks. In addition to EigenTrust [7] another piece of work that inspires this work is PeerTrut [18]. PeerTrust is the first to introduce the feedback credibility to aggregate the global reputation. For each rating peer, there is a corresponding rating credibility. In this way the ratings of malicious peer can be constrained. RLM [14] proposes Kalman feedback aggregation to adjust the affection of a malicious feedback through the parameter of estimated feedback variance. We have presented the design of EigenTrust^{++}, and showed analytically and experimentally that EigenTrust^{++} is significantly more attack resilient to EigenTrust while preserving all the best features of EigenTrust. Concretely, EigenTrust^{++} promotes three principled design goals. First, the normalized local trust value of a participant in a network should be computed by taking into account both quality (satisfactory and unsatisfactory experiences) and quantity of its interactions with other participants in the network, making it harder and costly for attackers to manipulate the trust model. Second, EigenTrust^{++} advocates a clean separation of the transaction or interaction quality from the feedback quality in reputation computation. This significantly strengthens the attack resilience of EigenTrust^{++}. Third but not the least, EigenTrust^{++} enables the propagating trust computation to incorporate weighted probabilistic iteration, instead of uniform probability distribution in EigenTrust. This design principle can capture the non-uniformity of trust propagation and experience sharing among the circle of "friends" (connected nodes in the network). We conduct extensive experimental evaluation on EigenTrust^{++}, and show that EigenTrust^{++} can significantly outperform EigenTrust in terms of both performance and attack resilience in the presence of dishonest feedbacks and sparse network connectivity against four representative attack models.

ACKNOWLEDGMENT

This work is partially funded by grants from NSF CISE NetSE program, NSF CISE CrossCutting program, an IBM faculty award, a grant from Intel ICST on Cloud Computing, National Nature Science Foundation of China No.: 61272173, Graduate Creative Talents Project of DUT. The first author performed this work while he is visiting DiSL, School of CS in Georgia Institute of Technology, funded by China Scholarship Council.

REFERENCES

[1] B. Edmonds, E. Norling, and D. Hales. Towards the evolution of social structure. *Computational & Mathematical Organization Theory*, 15(2):78–94, 2009.

[2] Q. Feng, L. Liu, and Y. Dai. Vulnerabilities and countermeasures in context-aware social rating services. *ACM Transactions on Internet Technology (TOIT)*, 11(3), 2012.

[3] J. Golbeck. Weaving a web of trust. *Science*, 321(5896):1640–1641, 2008.

[4] A. Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, 2001.

[5] A. Jøsang and R. Ismail. The beta reputation system. In *Proceedings of the 15th Bled Electronic Commerce Conference*, volume 160, 2002.

[6] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[7] 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*, pages 640–651. ACM, 2003.

[8] A. Marcozzi and D. Hales. Emergent social rationality in a peer-to-peer system. *Advances in Complex Systems (ACS)*, 11(04):581–595, 2008.

[9] T. Moscibroda, S. Schmid, and R. Wattenhofer. On the topologies formed by selfish peers. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 133–142. ACM, 2006.

[10] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

[11] S. Schmidt, R. Steele, T.S. Dillon, and E. Chang. Fuzzy trust evaluation and credibility development in multi-agent systems. *Applied Soft Computing*, 7(2):492–505, 2007.

[12] TM/RM simulator:. http://rtg.cis.upenn.edu/qtm/p2psim.php3.

[13] S. Song, K. Hwang, R. Zhou, and Y.K. Kwok. Trusted p2p transactions with fuzzy reputation aggregation. *Internet Computing, IEEE*, 9(6):24–34, 2005.

[14] X. Wang, L. Liu, and J. Su. Rlm: A general model for trust representation and aggregation. *IEEE Transactions on Services Computing*, 5(1):131–143, 2012.

[15] Y. Wang, V. Cahill, E. Gray, C. Harris, and L. Liao. Bayesian network based trust management. In *Autonomic and trusted computing: third international conference, ATC 2006, Wuhan, China, September 3-6, 2006: proceedings*, volume 4158, page 246. Springer-Verlag New York Inc, 2006.

[16] Y. Wang and A. Nakao. On cooperative and efficient overlay network evolution based on a group selection pattern. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 40(2):493–504, 2010.

[17] Y. Wang and A. Nakao. Poisonedwater: An improved approach for accurate reputation ranking in p2p networks. *Future Generation Computer Systems*, 26(8):1317–1326, 2010.

[18] L. Xiong and L. Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.