Leveraging Social Networks to Combat Collusion in Reputation Systems for Peer-to-Peer Networks

Ze Li, *Student Member, IEEE,* Haiying Shen*, *Member, IEEE*, Karan Sapra

Abstract—In peer-to-peer networks (P2Ps), many autonomous peers without preexisting trust relationships share resources with each other. Due to their open environment, the P2Ps usually employ reputation systems to provide guidance in selecting trustworthy resource providers for high reliability and security; however node collusion impairs the effectiveness of reputation systems in trustworthy node selection. Although some reputation systems have certain mechanisms to counter collusion, the effectiveness of the mechanisms is not sufficiently high. In this paper, we leverage social networks to enhance the capability of reputation systems in combating collusion. We first analyzed real trace of the reputation system in the Overstock online auction platform which incorporates a social network. The analysis reveals the impact of the social network on user purchasing and reputation rating patterns. We thus identified suspicious collusion behavior patterns and propose a social network based mechanism, SocialTrust, to counter collusion. SocialTrust adaptively adjusts the weight of ratings based on the social distance and interest relationship between peers. Experiment results show that SocialTrust can significantly strengthen the capability of current reputation systems in combating collusion.

Index Terms—Peer to peer networks, Reputation systems, Collusion, Social networks.

1 Introduction

The past decade has seen a rapid development of peer-to-peer networks (P2Ps) along with a dramatic surge of applications including file sharing (e.g., BitTorrent [1]), video streaming (e.g., PPLive [2]), and computing resource sharing (e.g., MAAN [3]). In these P2P applications, peers (acquaintance and non-acquaintance) directly contact each other to conduct transactions on resources (e.g., files, videos and etc.). Considering P2Ps' open environment where many autonomous nodes share resources, a critical problem is how a resource requester can choose a resource provider that is trustworthy and provides high-quality of service (QoS).

To deal with this problem, P2Ps usually employ reputation systems for reliability and security. Like the reputation systems in eBay [4], Amazon [5] and Overstock [6], a reputation system employed in P2Ps computes and publishes global reputation value for each node based on a collection of local ratings from other users in order to provide guidance in selecting trustworthy nodes; however reputation systems are generally vulnerable to node collusion [7], [8], which impairs their effectiveness in trustworthy service selection. A colluding collective is a group of malicious peers who know each other, give each other high ratings, and give all other peers low ratings in an attempt to subvert the system and gain high global reputation values [9].

A number of reputation systems employ certain mechanisms to fight against collusion. Although the mechanisms can reduce the influence of collusion on reputations to a certain extent, they are not sufficiently effective in countering collusion, or they contradict the P2Ps' goal of global resource sharing. Some mechanisms assign higher weights to ratings from pretrusted peers and (or) assign weights to ratings according to the raters' global reputations [10]–[12]. However, colluders can rate each other in a high frequency or compromise pretrusted peers to quickly raise their reputations. In other mechanisms, a peer evaluates others' trustworthiness based on the experience [13]–[15] of itself or its friends [16]. However, these mechanisms limit the service options and prevent strangers from freely conducting transactions between each other.

Due to the soaring popularity of the online social network (e.g., Facebook), more and more applications (e.g. Overstock [6], Oneswarm [17]) incorporate online social networks into their services to increase user interactions. In this paper, we propose a mechanism called SocialTrust that leverages social networks to enhance the effectiveness of current mechanisms in combating collusion in P2P networks. SocialTrust works for a P2P network integrated with an online social network (already exists or newly constructed) such as the Maze file sharing system [7]. In a reputation system, after a client's resource/service request is resolved by a server, the client rates the service quality of the server. The reputation system collects all ratings of a node and calculates its global reputation value, which is used to guide subsequent server selection. SocialTrust is built upon the reputation system of the P2P network and re-scales node reputation values based on user social information to mitigate the adverse influence of collusion. To investigate the impact of a social network on user purchasing and rating patterns and define the suspicious behaviors in P2P network, we analyzed a real trace of 450,000 transaction ratings during 2008-2010 that we crawled from Overstock Auctions (Overstock in short) [6]. Overstock is an online auction platform similar to eBay, but it distinguishes itself by integrating a social network into the market community. We found that *social closeness* and *interest similarity* impact user purchasing and rating patterns. First, users tend to buy products from high-reputed users or socially-close (3 hops or less) users. They also rate socially-close users with high ratings. Second, 88% of a user's purchases is within 20% of the user's product interest categories

- * *Corresponding Author. Email: shenh@clemson.edu; Phone: (864) 656 5931; Fax: (864) 656 5910.*
- *The authors are with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, 29634. E-mail: {zel, shenh, ksapra}@clemson.edu*

Digital Object Indentifier 10.1109/TC.2012.158

on average, and 60% of transactions are conducted between users sharing >30% interest similarity.

The observations on the purchasing transactions in Overstock can be directly mapped to resource transactions in P2P applications, in which a peer selects a server for a resource/service request. Based on our observations, we identified suspicious collusion behavior patterns based on the distance and interest relationship between peers in a social network. SocialTrust adjusts the ratings of suspicious colluders according to the social closeness and interest similarities of nodes in order to reduce the impact of collusion. By preventing colluders from gaining profits (e.g., reputations value) through collusion, the colluders underlying business model will be destroyed. Then nodes do not have incentives to collude with each other.

This work is the first that leverages a social network to identify suspicious collusion behavior patterns and reduce the influence of collusion on reputation systems. In summary, this work makes the contributions below:

(1) We crawled and analyzed user transaction trace from Overstock and found that buyer purchasing and rating behaviors are greatly affected by the distance and interest similarity of users in the social network and by seller reputation. Accordingly, we identified a number of suspicious collusion behavior patterns.

(2) We propose the SocialTrust mechanism to enhance a reputation system's capability in countering suspicious collusion behaviors learned from the trace data. SocialTrust adjusts the ratings from suspected colluders based on rater-ratee social closeness and interest similarity.

(3) We conducted extensive experiments to evaluate SocialTrust's effectiveness in handling different types of collusions. The experimental results show that current reputation systems are not sufficiently effective in dealing with collusion, and SocialTrust can significantly enhance their capability to effectively counter collusion.

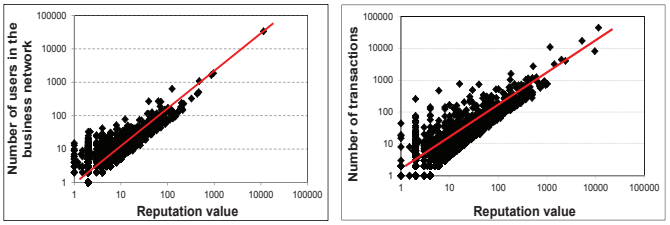
The remainder of this paper is as follow. Section 2 introduces related works in reputation systems and in collusion deterrence. Section 3 presents our investigation on the real trace. Section 4 describes SocialTrust in detail. Section 5 presents the performance evaluation of SocialTrust. Section 6 concludes the paper with remarks on our future work.

2 Related Work

Reputation systems In recent years, many reputation systems [10], [12], [18]–[27] have been proposed. PeerTrust [18] computes peer reputation scores based on three basic trust parameters and two adaptive factors. Trustme [19] offers an approach toward anonymous trust management, which can provide mutual anonymity for both the trust host and the trust querying peer. EigenTrust [10] and PowerTrust [20] depend on the distributed hash tables to collect reputation ratings and calculate the global reputation value of each peer. TrustGuard [12] incorporates historical reputations and behavioral fluctuations of nodes into the estimation of their trustworthiness. FuzzyTrust [21] uses fuzzy logic inferences to better handle uncertainty, fuzziness, and incomplete information in peer trust reports. GossipTrust [22] enables peers to share weighted local trust scores with randomly selected neighbors until reaching global consensus on peer reputations. Scrubber [23] fights polluted file content by rating both the file provider and file. Credence [24] gives users a robust estimate of file authenticity (the degree to which an object's content matches its advertised description). Fabrizio *et al.* [25] proposed an approach to P2P security that enables each client to compute a personalized, rather than global, performance score for peers, and also distinguish peer performance from peer credibility. Both XRep [26] and X^2Rep [27] extend the work in [25] by additionally computing object reputations based on weighted peer voting.

Collusion deterrence Recently, a number of research works have been conducted on the problem of collusion in reputation systems. EigenTrust [10] breaks collusion collectives by assigning higher weight to the feedback of pretrusted peers. Moreton *et al.* [28] proposed the Stamp algorithm, where peers issue stamps as virtual currency for each interaction, and the value of each peer's stamps is maintained by exchange rates that act as reputation values. TrustGuard [12] gives more weight to the feedbacks from similar ratings, acting as an effective defense against potential collusive nodes that only give good ratings within the clique and give bad rating to everyone else. Qiao *et al.* [7] analyzed the traffic logs in a P2P file sharing system to study different types of collusion patterns. Mao *et al.* [11] introduced using social networks in the Maze P2P file sharing system to reduce the impact of collusion. The authors assumed that the pretrusted peers only trust their friends, and proved that the friend network of the pretrusted peers can help to detect colluders. The works in [13]–[15] let a peer evaluate others' trustworthiness based on its experience. Sorcery [16] lets clients utilize the overlapping voting histories of both their friends and the content providers to judge whether a content provider is a colluder. However, the social network based method limits the service options and constrains resource sharing to only between friends. It also cannot provide a global reputation of each node calculated by ratings from a variety of users to accurately reflect its trustworthiness. Our proposed method is the first that leverages social distance and interest relationship from a social network to identify suspicious collusion and to reduce its influence on node reputation.

Sybil attack deterrence Collusion shares similarity to Sybil attacks in the sense of forming a collective to gain fraudulent benefits. Thus, we also include a number of social network based works on Sybil attacks as related works. Since malicious users can create many identities but few trust relationships, there is a disproportionately-small "cut" in the graph between the Sybil nodes and the honest nodes. SybilGuard [29] exploits this property to bound the number of identities a malicious user can create. SybilLimit [30] improves SybilGuard by leveraging multiple independent instances of the random route protocol to perform many short random routes and exploiting intersections on edges instead of nodes. SybilInfer [31] builds a probabilistic model of honest social networks and a Bayesian inference engine that returns potential regions of dishonest nodes. SumUp [32] prevents Sybil attack by comparing the social structure of colluders with that of non-colluders. Viswanath and Post [33] compared the existing designs of the Sybil defense schemes and showed that some community detection algorithms can defend against Sybil attacks.



(a) Business network size vs. reputation of each user (C=0.996) (b) # of transactions vs. reputation of each user

Fig. 1: Effect of reputation on transaction.

They also demonstrated that a well-defined community structure is inherently more vulnerable to Sybil attacks. Lesniewski-Laas *et al.* [34], [35] proposed a Sybil-resilient distributed hash table routing protocol to reduce the probability of routing collusion. These schemes can be used to complement SocialTrust to strengthen its capability to detect collusion behaviors.

3 Analysis of Real Trace in Overstock

In order to study the relationship between a user social network, transaction, and reputation system, we analyzed our crawled data of 450,000 transactions between over 200,000 users from Sep. 1, 2008 to Sep. 1, 2010 in Overstock. Overstock is an online e-commerce website that provides an online auction platform to a large community of users worldwide to conduct P2P e-commerce. A buyer and a seller on Overstock rate each other after a transaction, and the ratings are aggregated to form a user's global reputation. The range of ratings in Overstock is [-2,+2]. Each user has a "personal network" and a "business network." The "personal network" is a social network that is comprised of users connected by friendship links. If a user accepts the friend invitation from another user, a friendship link is established between the two users in the social network. A user can list hobbies and interests, post photos, and publish friend and business contact lists in his/her personal page in the personal network. The "business network" records the user's business contact list. Every time after a user completes a transaction, (s)he adds the transaction partner into his/her business network.

To crawl the data, we first selected a user in the Overstock as a seed node, and then used the breadth first search method to search through each node in the friend list in the personal network and business contact list in the business network. Based on the trace data, we try to identify suspicious collusion behavior patterns based on two main characteristics of collusion described in [7], [10]. First, colluders are normally socially-close nodes. Second, colluders frequently rate each other with high values in order to boost the reputation values of each other and (or) give others low values in order to suppress their reputation values and gain benefits.

3.1 Relationship between reputation, social network and transaction

Since users usually refer to sellers' reputations for seller selection, we first investigated the relationship between a user's reputation and the number of users in the user's business network. Figure 1(a) shows that there is a linear relationship between the reputation value of a user and the size of the user's business network. The strength of the linear association between two variables, x and y, can be quantified by the correlation coefficient, $C = s_{xy}^2/s_{xx}s_{yy}$, where $s_{xy} = \sum(x_i - \bar{x})(y_i - \bar{y})$, $s_{xx} = \sum(x_i - \bar{x})^2$ and $s_{yy} = \sum(y_i - \bar{y})^2$. The correlation coefficient between the reputation value and business network size is 0.996. Since users prefer to buy products from trustworthy users, sellers with higher reputations are more likely to attract more buyers, hence have larger business networks. This is confirmed by Figure 1(b), which shows the number of transactions a user has received is proportional to his/her reputation. This means that users with higher reputations attract more transactions. This is also the motivation of colluders to conspire together to boost the reputation of each other. Thus, we make an observation (O) from the results:

O1: Users with higher reputation values are more likely to attract more buyers, and users seldom buy products from low-reputed sellers.

We then derive an inference (I) from O1.

I1: A buyer is unlikely to frequently rate a low-reputed user with high or low ratings, since (s)he is unlikely to repeatedly choose a seller with low QoS.

Figure 2 shows the number of users in the personal network of each user versus her/his reputation value. We can see that there is a very weak linear relationship between personal network size and reputation value. Their correlation coefficient is only 0.092. The linear relationship may be caused by the reason that a high-reputed user knows many users from his/her large business network, who may become the user's friends. The weak linear relationship implies that a low-reputed user may have the same personal network size as a high-reputed user.

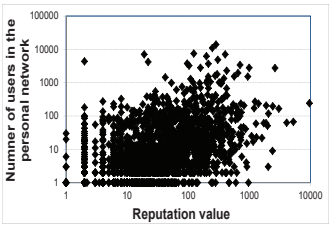


Fig. 2: Social network size vs. reputation (C=0.092)

O2: A low-reputed user may have a large number of friends in his/her social network.

I2: A low-reputed user may have many socially-close friends that (s)he can collude with in order to increase his/her reputation.

3.2 Impact of social closeness

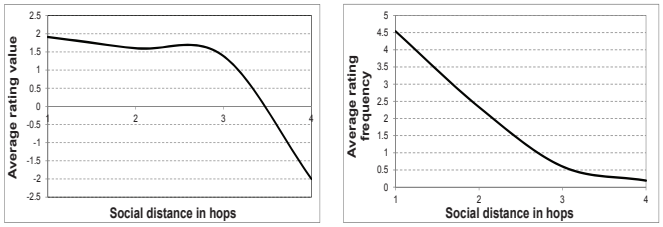
Social distance between two nodes is the number of hops in the shortest path between them in the personal network, which represents the social closeness between the two users. If two users are directly connected in the personal network, their social distance is 1. Next, we investigate the impact of social distance on user purchasing and reputation rating behavior.

Figures 3(a) and (b) show the average rating values and average number of ratings from buyers to sellers with different social distances in hops ≤ 4, respectively. We see that as the social distance between people increases, the average rating values and average number of ratings decrease.

O3: Most transactions with high ratings occur between users within 3 hops.

Thus, we identify a suspicious behavior (B) of collusion:

B1: Users with long social distances rate each other with high ratings and high frequency.



(a) Ave. value of ratings. (b) Ave. # of ratings.

Fig. 3: Impact of social distance on reputation and transaction.

O4: Users with shorter social distances are more likely to rate each other with higher ratings and higher frequency.

From I1, I2 and O4, we retrieve:

B2: A user frequently rates a low-reputed socially-close user with high ratings.

3.3 Impact of social interest similarity

Next, we investigate the impact of user interest on user purchasing patterns. We classified the products bought or sold by the users into categories such as Electronics, Computers, and Clothing. We then generated an interest set $\mathcal{V}$=$<v_1, v_2, v_3, ..., v_k>$ for each user, where v denotes a product category. We ranked the categories that each buyer has purchased from in descending order of the number of the products (s)he has purchased in each category. We define the *percent of a category rank* as the ratio of the average number of products in the category rank per user over the average number of all products bought per user. Figure 4(a) plots the Cumulative Distribution Function (CDF) of the percent of each category rank. The figure shows that the number of products in different category ranks conforms to a power law distribution. It also shows that the top 3 categories of products constitute about 88% of the total number of products a user bought. Thus,

O5: A user mostly buys products in a few categories ($\leq$3) in which (s)he is interested.

It was indicated that normal nodes primarily request items in their interests [36]. Our above analytical results are consistent with this finding. We calculated the interest similarity between each pair of buyer n_i and seller n_j by

$$\frac{|\mathcal{V}_i \cap \mathcal{V}_j|}{\min(|\mathcal{V}_i|, |\mathcal{V}_j|)}. \quad (1)$$

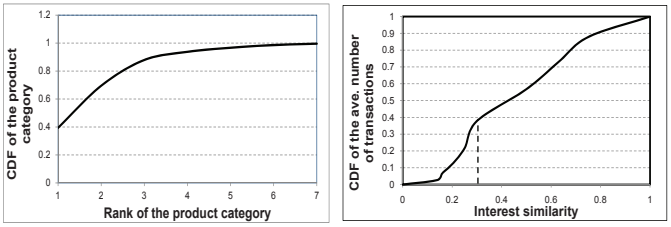
Figure 4(b) depicts the CDF of the average number of transactions versus interest similarity. We see only 10% of transactions are conducted between users with $\leq$20% interest similarity, 60% of transactions are conducted between users with >30% interest similarity, and more transactions occur between users with interest similarity higher than 50%.

O6: A buyer seldom buys products from sellers with low interest similarity.

B3: Users with few common-interests rate each other with high ratings and high frequency.

Based on O1, I1 and O6, we know that a seller may try to suppress the reputation of his/her competitors who sell similar products by frequently rating them with low ratings. Thus, we identify another suspicious behavior:

B4: A buyer frequently rates a seller with many common-interests with low ratings.



(a) CDF of the top 7 category ranks. (b) Ave. number of transactions over all pairs of users.

Fig. 4: Impact of interests on purchasing patterns.

4 SocialTrust: Social Network based Mechanism to Combat Collusion

SocialTrust can be used in any reputation system for P2P networks to enhance its capacity to combat collusion. As most reputation systems for P2P networks [10], [12], [18]–[27], we assume that most of the users in the network are rational and legitimate nodes. If a P2P network already has a social network like Overstock and the Maze file sharing system [7], SocialTrust can directly use the social network. Otherwise, SocialTrust provides a plugin for the social network construction. It requires users to enter their interest information and establish friend relationships as in other reputation systems [11], [16], [29]. Like current online social networks (e.g., Overstock and Facebook), SocialTrust maintains a record of interactions among users on the personal network. It processes both the interest information and interaction information for combating collusion.

SocialTrust derives the *social closeness* (from the social relationship and node interaction) and *interest similarity* (from node profiles or activities) between a pair of nodes. We use Ω_c and Ω_s to respectively denote these two coefficients. SocialTrust detects action patterns of suspicious collusion behaviors and then reduces the weight of the ratings from suspected colluders based on the two coefficients. Note that some non-colluders' behaviors might be coincident with our identified suspicious behavior patterns. However, such cases are rare since the suspicious behaviors are abnormal as shown in Section 3. Since collusion from malicious nodes greatly impairs the system performance, the benefits from reducing collusion's influence should outweigh the effect of a marginal amount of possible false positives in SocialTrust.

4.1 Social closeness based collusion deterrence

Social network studies [37], [38] indicate that the number of social relationships and interaction frequency determine the social closeness of a pair of adjacent nodes, n_i and n_j. The social relationships (e.g., colleague and classmate) between a pair of nodes are always indicated in a social network, and their interaction frequency means their online contact frequency. In a P2P network incorporated with a social network, an interaction can be regarded as an action that a peer requests a resource from another peer. More relationships between two nodes mean a closer relationship between them. Also, if n_i interacts with n_j more frequently than with other friends, it means that n_i is socially-closer to n_j. Rather than using the complex supervised learning algorithms [37], [38] for social closeness modeling, which is computationally expensive especially in a large-scale distributed P2P system with dynamic interaction frequency

and social relationship information updates, SocialTrust introduces a lightweight social closeness model.

In SocialTrust, considering the two factors, the social closeness $\Omega_{c_{(i,j)}}$ between two adjacent nodes n_i and n_j is calculated by

$$\Omega_{c_{(i,j)}} = \frac{m_{(i,j)} f_{(i,j)}}{\sum_{k=0}^{|\mathcal{S}_i|} f_{(i,k)}}, \tag{2}$$

where $m_{(i,j)} \geq 1$ denotes the number of social relationships between n_i and n_j, $f_{(i,j)}$ denotes the social interaction frequency from n_i to n_j, and $\mathcal{S}_i$ denotes a set of neighbors of node i, where $|\mathcal{S}_i|$ denotes the number of nodes in $\mathcal{S}_i$.

Intuitively, for a pair of non-adjacent nodes that rate each other but have no direct social relationship, fewer hops in the shortest path between the two nodes in the social network graph means a closer relationship. Since each node establishes its own friend-relationship network, broadcasting can be used to find the shortest paths. However, broadcasting generates a large amount of overhead. Binzel *et al.* [39] indicates that a reduction in social distance between two people significantly increases the trust between them. Also, the trace data from Overstock shows that users normally do business with others within 3 hops in their personal networks, which is consistent with the observation in [40] that the users possessing a social network primarily transact with 2 to 3 hop partners. Therefore, the friend-of-friend (FOF) relationship [41] is sufficiently accurate to capture the indirect social closeness between two nodes. Meanwhile, if two nodes have more common friends, they are more likely to have a close social relationship based on the homophily feature in social science [42]. Therefore, using $\mathcal{S}_i$ and $\mathcal{S}_j$ to respectively denote the set of friends of two non-adjacent nodes, n_i and n_j, the social closeness between two non-adjacent nodes n_i and n_j is:

$$\Omega_{c_{(i,j)}} = \sum_{k \in |S_i \cap S_j|} \frac{\Omega_{c_{(i,k)}} + \Omega_{c_{(k,j)}}}{2} \tag{3}$$

That is, we find all the common friend n_k between node n_i and n_j. The social closeness between n_i and n_j through n_k is calculated by averaging the closeness of $\Omega_{(i,k)}$ and $\Omega_{(k,j)}$ if the common friends exsit. Otherwise, the closeness value is the minimum social closeness value of neighbor nodes in the social paths between n_i and n_j [43]. In summary:

$$\Omega_{c_{(i,j)}} = \begin{cases} \frac{m_{(i,j)} \cdot f_{(i,j)}}{\sum_{k=0}^{|\mathcal{S}_i|} f_{(i,k)}} & \text{adjacent nodes,} \\ \sum_{k \in |S_i \cap S_j|} \frac{\Omega_{c_{(i,k)}} + \Omega_{c_{(k,j)}}}{2} & \text{non-adjacent nodes, } k \neq \emptyset \\ \min_{1 \leq i \leq n} \Omega_{c_{(k_i, k_{i+1})}} & \text{non-adjacent nodes, } k = \emptyset, \end{cases} \tag{4}$$

where node k_i is in the path between n_i and n_j.

Suppose $\bar{F}$ is the average rating frequency from one node to another node in the system, SocialTrust uses $\theta\bar{F}$ $(\theta > 1)$ as the threshold to determine whether the rating frequency is high, where θ is a scaling parameter. For example, in Overstock, $\bar{F} = 2.2$/month. According to *B3* and *B4* described in Section 3, when n_i rates n_j with high ratings and high frequency, if $\Omega_{c_{(i,j)}}$ is very low or very high and n_j's reputation is low, it means n_i is potentially a colluder. Then, SocialTrust reduces the weight of the ratings from n_i to n_j based on $\Omega_{c_{(i,j)}}$. For this purpose, SocialTrust relies on Gaussian function as a reputation filter.

As shown in Figure 5, the Gaussian function has a characteristic symmetric "bell curve" shape that can mitigate or filter the effect of a factor with values greatly deviated from the normal value. That is,

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}}, \tag{5}$$

where parameter a is the height of the curve's peak, b is the position of the centre of the peak, and c controls the width of the "bell". SocialTrust uses the Gaussian function to adjust the ratings from n_i to n_j, denoted by $r_{(i,j)}$.

$$r_{(i,j)} = r_{(i,j)} \cdot \alpha \cdot e^{-\frac{(\Omega_{c_{(i,j)}} - \bar{\Omega}_{c_i})^2}{2|\max \Omega_{c_i} - \min \Omega_{c_i}|^2}}, \tag{6}$$

where α is the function parameter. $\max \Omega_{c_i}$, $\min \Omega_{c_i}$, and $\bar{\Omega}_{c_i}$ respectively denote the maximum, minimum, and average social closenesses of n_i to the nodes that n_i has rated.

We set $\alpha = a$ to adjust the weight of ratings, $b = \bar{\Omega}_{c_i}$, which is the most reasonable social closeness of n_i to other nodes it has rated, and $c = |\max \Omega_{c_i} - \min \Omega_{c_i}|$, which is the greatest variance of social closeness of n_i to other nodes it has rated. The exponent in Equation (6) is the deviation of $\Omega_{c_{(i,j)}}$ from the normal social closeness of n_i to other nodes it has rated. We also can replace Ω_{c_i} with the average Ω_c of a pair of transaction peers in the system based on the empirical result.

As Figure 5 shows, the Gaussian function can significantly reduce the weights of the ratings from the nodes with very high or very low social closeness to the ratees, mildly reduce the weights of those from the nodes with high or low social closeness to the ratees, and nearly maintain the ratings from the nodes with normal closeness to the ratees. As a result, the weight from the ratings from suspected colluders is reduced.

4.2 Interest similarity based collusion deterrence

In SocialTrust, each node has an interest set $\mathcal{V}$=<$v_1, v_2, v_3, ..., v_k$> indicating its interests. In P2P applications, the interests of a peer can be derived from the resources it frequently requests or from the interests in the user's social network profile. As mentioned, the social interest similarity between n_i and n_j is calculated by:

$$\Omega_{s(i,j)} = \frac{|\mathcal{V}_i \cap \mathcal{V}_j|}{\min(|\mathcal{V}_i|, |\mathcal{V}_j|)}. \tag{7}$$

Nodes with larger Ω_s share more interests.

One property of social networks is that nodes with common interests tend to interact with each other more often than with other nodes [42]. This was confirmed in previous study [44] on peoples' relations based on their interested files. P2P resource sharing and transactions usually occur between nodes sharing similar interests, As a result, if two nodes n_i and n_j sharing few interests (i.e., small $\Omega_{s(i,j)}$) rate each other frequently, they are likely to be colluding with each other as indicated in *B3* in Section 3. As indicated in *B4*, if two nodes having a high interest similarity but one frequently rates the other with low ratings, they are likely to be business competitors and the rater is a potential malicious node.

In these two cases, SocialTrust reduces the weight of the ratings from suspected colluders that have very high or low $\Omega_{s(i,j)}$ with the ratee using the Gaussian function:

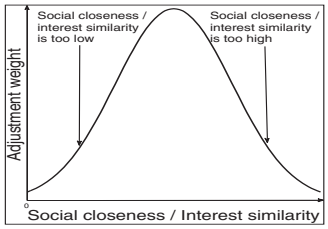


Fig. 5: One-dimensional reputation adjustment.

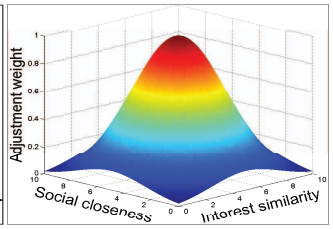


Fig. 6: Two-dimensional reputation adjustment.

$$r_{(i,j)} = r_{(i,j)} \cdot \alpha \cdot e^{-\frac{(\Omega_{s_{(i,j)}} - \bar{\Omega}_{s_i})^2}{2|\max \Omega_{s_i} - \min \Omega_{s_i}|^2}}, \quad (8)$$

where $\max \Omega_{s_i}$, $\min \Omega_{s_i}$ and $\bar{\Omega}_{s_i}$ denote the maximum, minimum and average interest similarity of node n_i with the nodes it has rated, respectively. According to *B3* and *B4*, the rating from n_i to n_j is adjusted according to Equation (8) when n_i frequently rates n_j with high ratings and $(\Omega_{s_{(i,j)}} - \bar{\Omega}_{s_i}) < 0$ which implies that n_i and n_j share few interests, or when n_i frequently rates n_j with low ratings and $(\Omega_{s_{(i,j)}} - \bar{\Omega}_{s_i}) > 0$ which implies that n_i and n_j share many interests.

Similar to social closeness, we also can replace Ω_{s_i} with the average Ω_s of a pair of transaction peers in the system based on the empirical result. For example, in Overstock, the average, maximum and minimum interest similarity between a pair transaction peers are 0.423, 1, and 0.13.

4.3 Combined social closeness and similarity based collusion deterrence

Combining Formulas (6) and (8), we get:

$$r_{(i,j)}(\Omega_c, \Omega_s) = r_{(i,j)} \cdot \alpha \cdot e^{-(\frac{(\Omega_{c_{(i,j)}} - \bar{\Omega}_{c_i})^2}{2|\max \Omega_{c_i} - \min \Omega_{c_i}|^2} + \frac{(\Omega_{s_{(i,j)}} - \bar{\Omega}_{s_i})^2}{2|\max \Omega_{s_i} - \min \Omega_{s_i}|^2})}, \quad (9)$$

which simultaneously considers social closeness and interest similarity. For example, if two low-reputed nodes rating each other with high frequency have a very close social relationship (i.e., high $\Omega_{c_{(i,j)}}$) but share few common interests (i.e. low $\Omega_{s_{(i,j)}}$), they are more likely to collude with each other. This is because two nodes have low probability to frequently request resources from each other if they share few common interests, and it is unlikely that a node will request a resource from a low-reputed node. Let us use H_c and L_c to denote very high and low social closeness and use H_s and L_s to denote very high and low interest similarity. As Figure 6 shows, the rating values between the nodes that have (H_c, H_s), (H_c, L_s), (L_c, H_s), and (L_c, L_s) are greatly reduced. Therefore, based on Formula (9), the influences of the collusion listed in *B1-B4* are reduced.

SocialTrust can be executed in a centralized or distributed manner. In the centralized system, a centralized reputation manager manages all node reputation values and social information, and adjusts the values based on social closeness and similarity. As a distributed SocialTrust system is more challenging and suitable for large-scale distributed P2P networks, here, we introduce how SocialTrust is executed in a distributed manner. It can be easily adapt to the centralized reputation system.

In a reputation system, one or a number of trustworthy node(s) function as resource manager(s). Each resource manager is responsible for collecting the ratings and calculating the global reputation of certain nodes. Thus, each resource manager can keep track of the rating frequencies and values of other nodes for the nodes it manages, which helps them to detect collusion in SocialTrust. A manager adjusts the ratings from suspected colluders when periodically calculating node global reputation. Suppose M_j is the resource manager of n_j. M_j keeps the interest set and friendlist of n_j. After each reputation update interval T, M_j calculates the number of positive and negative ratings during T from each rater node n_i for n_j, denoted by $t^+_{(i,j)}$ and $t^-_{(i,j)}$.

SocialTrust sets the thresholds for positive rating frequency and negative rating frequency of a node, denoted by T^+_t and T^-_t from empirical experience. For example, in Overstock, the average, maximum and minimum numbers of positive ratings of a node per month are 1.75, 21 and 1, while those of negative ratings are 1.84, 2 and 1. When $t^+_{(i,j)} > T^+_t$ or $t^-_{(i,j)} > T^-_t$, which means that n_i is a suspected colluder, and M_j does not have interest set and friendlist of rater n_i, it contacts n_i's reputation manager M_i for the information. Based on the calculated $\Omega_{c_{(i,j)}}$ and $\Omega_{s_{(i,j)}}$ and n_j's reputation, M_i makes further judgement and adjusts the $r_{(i,j)}$ accordingly.

Specifically, SocialTrust sets a threshold for global reputation (R) of a low-reputed node, denoted by T_R. It also sets high and low thresholds for $\Omega_{c_{(i,j)}}$ and $\Omega_{s_{(i,j)}}$ to represent the degree of social closeness and interest similarity between a pair of nodes, denoted by T_{c_h}, T_{c_l}, T_{s_h}, and T_{s_l}, respectively. If $t^+_{(j,i)} > T^+_t$, which means n_j also frequently rates n_i with positive ratings, if (1) their social closeness is low ($\Omega_{c_{(i,j)}} < T_{c_l}$) (*B1*), (2) their social closeness is high ($\Omega_{c_{(i,j)}} > T_{c_h}$) and n_j is a low-reputed node ($R_j < T_R$) (*B2*), or (3) their interest similarity is low ($\Omega_{s_{(i,j)}} < T_{s_l}$) (*B3*), M_i adjusts $r_{(i,j)}$ according to Equation (9). If $t^-_{(i,j)} > T^-_t$, which means n_i frequently rates n_j with negative ratings, and their interest similarity is high ($\Omega_{s_{(i,j)}} > T_{s_l}$) (*B4*), M_i adjusts $r_{(i,j)}$.

4.4 Resilience to falsified static social information

Since SocialTrust depends on the social network to combat collusion, colluders may manipulate the social network to counterattack SocialTrust. Recall that SocialTrust identifies reputation raters and ratees with very high or very low social closeness or interest similarity. Thus, colluders would fabricate static social network information (i.e., social relationships and interests in profiles) to keep their social closeness and interest similarity at a moderate level in order to avoid being detected. In order to strengthen the ability of SocialTrust in combating collusion, we further improve the social closeness and interest similarity measurement's accuracy. We consider weight for the closeness of different social relationships. For example, kinship relationship should have higher weight than the friendship relationship. The social closeness between two nodes in Formula (3) is updated to:

$$\Omega_{c_{(i,j)}} = \begin{cases} \frac{(\sum_l \lambda^{(l-1)} w_{d_l}) \cdot f_{(i,j)}}{\sum_{k=0}^{|\mathcal{S}_i|} f_{(i,k)}}, & n_i \text{ and } n_j \text{ are adjacent,} \\ \sum_{k \in |S_i \cap S_j|} \frac{\Omega_{c_{(i,k)}} + \Omega_{c_{(k,j)}}}{2} & \text{non-adjacent nodes, } k \neq \emptyset, \\ \min_{1 \leq i \leq n} \Omega_{c_{(k_i, k_{i+1})}} & \text{non-adjacent nodes, } k = \emptyset, \end{cases} \quad (10)$$

where w_{d_l} denotes the weight of the l^{th} social relationship between n_i and n_j in the relationship list sorted

in descending order of the relationship weight, and $\lambda \in [0.5, 1]$ denotes a constant relationship scaling weight.

We also consider the weight of each node's interest. We use $w_{s_{(i,l)}}$ to denote the weight of node n_i on interest l. It equals the percent of node n_i's requests on interest l in all of its requests. Then, the interest similarity in Formula (11) is updated to:

$$\Omega_{s_{(i,j)}} = \frac{\sum_l w_{s_{(i,l)}} \cdot w_{s_{(j,l)}}}{\min(|\mathcal{V}_i|, |\mathcal{V}_j|)},\ l \in \{\mathcal{V}_i \cap \mathcal{V}_j\}. \quad (11)$$

Thus, SocialTrust depends not only on the static social network information but also the real node interaction in the social network and resource requests, which prevents the colluders from manipulating the social information.

To avoid being detected due to *B1*, colluders would try to increase their social closeness by increasing their total number of social relationships. As shown in Equation (10), the social closeness $\Omega_{c_{(i,j)}}$ is determined not only by the number of social relationships but also the interaction frequency. Although adding more social relationships can increase $\Omega_{c_{(i,j)}}$, the increment can be very small if the interaction frequency coefficient $f_{(i,j)}$ is low (i.e., the nodes actually are not socially close). To avoid being detected due to *B2*, colluders would try to reduce the number of social relationships to gain a median social closeness value. Similarly, a pair of nodes n_i and n_j with high interaction frequency $f_{(i,j)}$ (i.e., the two nodes are really socially-close) still have large social closeness value $\Omega_{c_{(i,j)}}$. As SocalTrust considers the weight of social closeness and exponentially decreases the effect of the closeness weight on the final social closeness metric, the calculated $\Omega_{c_{(i,j)}}$ can more accurately reflect the node social closeness. If two colluders add or reduce low-weight social relationships between them, it only slightly changes the social closeness.

To avoid being detected due to *B3*, colluders would try to increase their common-interests. The colluders may fill out false interest information in their profiles to match their colluders' interests to gain a reasonably high interest similarity value in SocialTrust. As shown in Equation (11), in addition to a node's interests, its percent of requests on the interests is also considered when calculating the interest similarity $\Omega_{s_{(i,j)}}$. Although a colluder's profile lists many interests matching the boosting colluder's interests, their social interest similarity $\Omega_{s_{(i,j)}}$ is still small if the colluder does not have many requests on the interests, which implies that it really has no such interests. To avoid being detected due to *B4*, a colluder may reduce many common interests in its profile with its ratee to gain a moderate interest similarity with the ratee. However, the colluder's frequent requests on the deleted common interests still reveals that these are its interests. As shown in Equation (11), a colluder's many requests on these interests still leads to high interest similarity value with its ratee.

In conclusion, it is difficult for colluders to manipulate social network's information to counterattack SocialTrust. Since the underlying business model of colluders is to gain benefits from collusion, by preventing colluders from gaining high reputations value through suspicious behaviors, the colluders' underlying business model will be destroyed. Then nodes should not have incentives to collude each other by using SocialTrust.

5 Performance Evaluation

5.1 Experimental setup

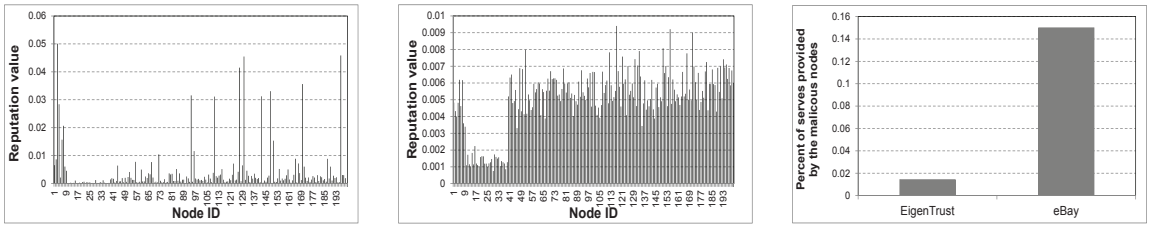
Network model. We built an unstructured P2P network with 200 nodes. According to the Overstock trace, the number of total interests in the P2P network was set to 20, and the number of interests for each node was randomly chosen from [1,10]. Nodes with the same interests are connected with each other, and a node requests resources (resource and service are interchangeable terms in this section) from its interest neighbors. In addition, we randomly assign [1-2] relationships between nodes in the system, and the colluders are randomly assigned with [3-5] relationships that have the same weight.

As observed in Section 3, in the experiments, the frequency at which a node requests resources in its interests conforms to a power law distribution. Each node can handle 50 requests simultaneously per query cycle. When selecting a service for its request, a node randomly chooses a neighbor with available capacity greater than 0 and reputation higher than $T_R = 0.01$.

Simulation execution. The simulation proceeds in simulation cycles. Each simulation cycle has 30 query cycles. In each query cycle, each peer issues a query if it is active. The probability that a node is active is randomly chosen from [0.5,1]. Among the colluders, the nodes receiving ratings from other nodes are called *boosted nodes*, and the nodes rating others are called *boosting nodes*. In each query cycle, a boosting node rates a boosted node [3-7] times on an interest randomly selected from the interests of the boosted node. Each experiment has 50 simulation cycles. Each experiment is run 5 times, and the average of the results is the final result. The 95% of the confidential interval is reported.

Node model. We consider three types of nodes: pretrusted nodes, malicious colluders, and normal nodes. The pretrusted and normal nodes always provide authentic resources to the requesters with a probability of 1 and 0.8, respectively. We use B to denote the probability that a malicious node offers an authentic file (i.e., good behavior). Since colluders usually offer low QoS [7], [8], we tested the performance of reputation systems when B=0.2 and 0.6, respectively. We randomly chose 9 pretrusted nodes and 30 colluders in the system. We assigned the social distance between colluders to 1. Considering most transactions in Overstock occur between nodes with 1-3 social distance, we set the social distances between all other nodes to values randomly chosen from [1,3]. In these experiments, we focus on the collusion behavior of B2 and B3. The social interaction frequency $f_{(i,j)}$ equals the rating frequency (i.e., resource transaction frequency) of n_i to n_j. In a nutshell, colluders have relatively more social relationships, higher social interaction frequency, and less common interests.

Collusion model. We consider the following three major collusion models [7]: the pair-wise collusion model (PCM), the multiple node collusion model (MCM), and the multiple and mutual collusion model (MMM). We consider positive ratings among colluders in the experiments. Similar results can be obtained for the collusion of negative ratings. In PCM, two colluders rate each other with a positive value at a high frequency in order to raise each other's reputation. In MCM, a



(a) Reputation distribution in EigenTrust. (b) Reputation distribution in eBay. (c) Percent of services provided by malicious nodes.

Fig. 7: Comparison of EigenTrust and eBay without colluders.

number of boosting nodes rate a single boosted node with high frequency in order to boost the reputation of that node, but the boosted node does not rate boosting nodes back. In MMM, a number of boosting nodes rate boosted nodes with high frequency, and the boosted nodes rate boosting nodes back.

Reputation model. The initial reputation of each node in the network is 0. A client gives a service rating of 1 when it receives an authentic service. Otherwise, the rating is -1. Each node's global reputation is updated once after each simulation cycle. The parameter α in the Gaussian function was set to 1. We measured the performance of the following three reputation systems: EigenTrust [10], eBay, and SocialTrust. We set the weight of reputations from pretrusted nodes in EigenTrust to 0.5. We use a simulation cycle to represent a week in eBay. After each simulation cycle, we scale the reputation of each node to [0,1] by $R_i/\sum_{k=0}^{n} R_k$, where R_i is accumulated ratings of n_i.

We assigned user IDs 1-9 to the pretrusted nodes and IDs 10-39 to the colluders. With a collusion-resilient reputation system, we expect to see that the nodes with ID 10-39 (i.e., colluders) have extremely low reputation values and the normal nodes have comparably higher reputation values. We also conducted experiments with different numbers of nodes and colluders. The relative performance differences between the different systems remain almost the same as those we will report. Though we determine the experimental setting parameters randomly in their reasonable ranges, changing the parameter values will not change the relative performance differences in a given experiment setup.

5.2 EigenTrust and eBay without colluders

The malicious nodes offer authentic files with probability randomly selected from [0.2, 0.6]. Figure 7(a) shows the reputation distribution of EigenTrust without colluders. It shows that the reputation values of the malicious nodes are very low and the reputation value of pretrusted node and a small number of normal nodes are comparatively high. This is because the node selection strategy in EigenTrust always retrieves files from the nodes with reputation larger than the threshold T_r. Since the pretrusted nodes and normal nodes offer more authentic files than malicious nodes, the reputation values of these nodes are much higher than malicious nodes.

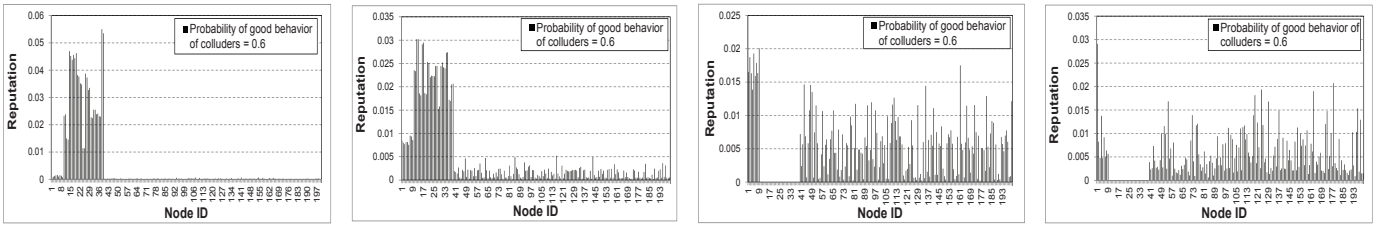
Figure 7(b) plots the reputation distribution of nodes in eBay without colluders. It shows the reputation values of the nodes are distributed relatively evenly and nodes with IDs in 10-39 have lower reputation. Since the malicious nodes have a high probability to send inauthentic files to other nodes, the reputation values of the malicious nodes are lower than other nodes. Nodes with lower reputation values attract less traffic requests. Since in eBay, a node's reputation increase is only determined by whether the node offers more authentic files than inauthentic files in each simulation cycle, the nodes with B>0.5 are possible to have good reputation values. In contrast, in EigenTrust, every misbehavior of a node is counted into the reputation calculation. This causes the reputation values of malicious nodes in EigenTrust to be lower than those of malicious nodes in eBay.

Figure 7(c) compares the percent of the service queries sent to malicious nodes in EigenTrust and eBay. It shows that the percent of the services provided by the malicious nodes in EigenTrust is much less than eBay. In eBay, since the reputation of a node is only based on whether it provides more authentic products than inauthentic products in a simulation cycle, the reputation updates in eBay are slow. Therefore, it takes a long time for a good node to gain a high reputation and for a malicious node to receive a low reputation. This allows a large amount of service queries to be sent to the malicious nodes.

5.3 Pair-wise collusion (PCM)

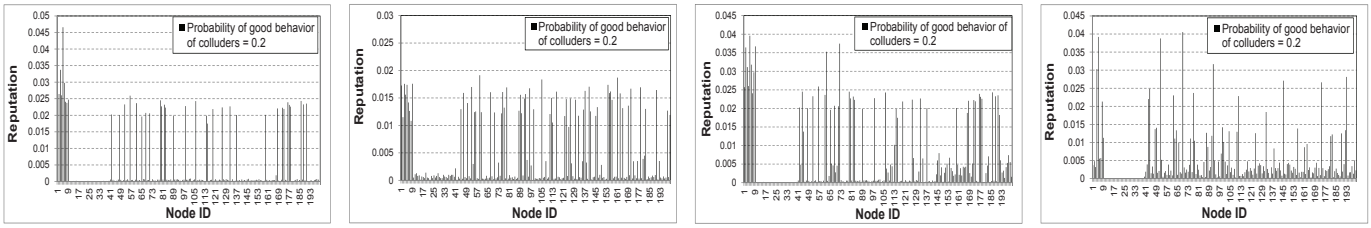
We first show the effectiveness of EigenTrust, eBay and SocialTrust in thwarting pair-wise collusion with colluders offering authentic services with 0.6 probability (B=0.6). The colluders rate each other with high frequency of 20 ratings per query cycle. Figure 8(a) shows the reputation distribution of all nodes in EigenTrust. We can see that colluders have much higher reputations than all other nodes. Also, the reputations of pretrusted nodes are slightly higher than normal nodes but are significantly lower than colluders. Since the colluders behave well with probability 0.6, they gain certain reputations. The colluders further increase the reputations of each other, which helps them to attract many service requests to further increase their reputations. Although the normal nodes and pretrusted nodes offer good services with probabilities of 0.8 and 1 respectively, their reputations are dramatically lower than colluders. Therefore, EigenTrust has low effectiveness in combating collusion, and its generated reputations cannot truly reflect the node trustworthiness when B=0.6. Figure 8(b) plots the reputation distribution of all nodes in eBay. It shows that the reputations of the colluders are much higher than all other nodes. The reason is that eBay enables the colluders with B=0.6 to increase reputations. Also, the mutual positive ratings between colluders further boost their own reputations. Therefore, eBay also has low effectiveness in combating collusion.

Comparing Figure 8(a) and Figure 8(b), the reputations of colluders in EigenTrust are higher than those in eBay, and the reputations of pretrusted and normal nodes in EigenTrust are much lower than those in eBay.



(a) EigenTrust. (b) eBay. (c) EigenTrust with SocialTrust. (d) eBay with SocialTrust.

Fig. 8: Reputation distribution in PCM with B=0.6 (pretrusted nodes: 1-9, colluders: 10-39).



(a) EigenTrust. (b) eBay. (c) EigenTrust with SocialTrust. (d) eBay with SocialTrust.

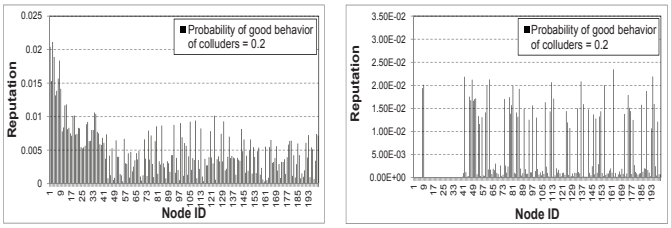
Fig. 9: Reputation distribution in PCM with B=0.2 (pretrusted nodes: 1-9, colluders: 10-39).

This is because in EigenTrust, the ratings from nodes are weighted based on the reputations of the nodes. Since the ratings from colluders with high reputation have high impact on the reputation calculation, the reputation values of the colluders can be quickly boosted. In eBay, the contribution of the ratings from the colluders is limited since no matter how frequently a node rates the other node in a simulation cycle, eBay only counts all the ratings as one rating. Therefore, eBay constrains reputation increase caused by collusion, leading to much lower reputations of colluders.

Figures 8(c) and (d) show the reputation distributions of the nodes in EigenTrust and eBay with SocialTrust, respectively. We can see that the colluders in both figures have much lower reputation values than those in Figures 8(a) and (b). The results show that SocialTrust can help EigenTrust and eBay to effectively thwart collusion. SocialTrust identifies suspected colluders based on social closeness and distance, and adjusts their reputation. This causes the colluders in SocialTrust to finally receive significantly low reputations. Since no nodes choose low-reputed nodes for services, SocialTrust effectively counters the collusion.

Next, we measure the reputation distribution of nodes when B=0.2 in different systems. Figure 9(a) shows the reputation distribution of nodes in EigenTrust. We see that EigenTrust is able to reduce the reputation values of the colluders. Although colluders rate each other frequently, the weight of their ratings is very low due to their low-QoS and reputations. Therefore, they finally receive low reputations and fewer service requests. Since the pretrusted nodes always behave well, they continuously receive high reputation values, finally gaining high reputations. We also notice that some normal nodes have high reputations while others have lower reputations. At the initial stage, a node randomly chooses from a number of options with the same reputation value 0. Since the chosen node earns reputation, it subsequently has a higher probability to be chosen. Therefore, EigenTrust can counter collusion when the colluders offer low-QoS.

Figure 9(b) shows the reputation distribution of nodes in eBay. The reputations of colluders are much lower



(a) EigenTrust. (b) EigenTrust with SocialTrust.

Fig. 10: Reputation distribution in PCM with compromised pretrusted nodes with B=0.2 (pretrusted nodes: 1-9, colluders: 10-39).

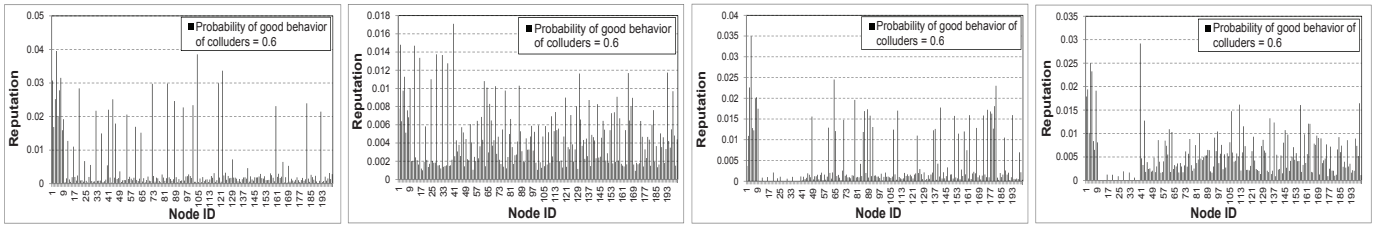
than those of the pretrusted nodes and normal nodes. The colluders receive low ratings from normal nodes due to their high probability of misbehavior. Although the colluders rate each other with high frequency to boost their reputations, eBay disregards the ratings from the same rater in the same simulation cycle, leading to a low final reputation. Because colluders still receive high unweighted ratings with probability of 0.2, they earn slightly higher reputations than in EigenTrust.

Figures 9(c) and (d) show the reputation distribution of nodes in EigenTrust and eBay with SocialTrust. Both figures show that the reputation values of colluders are nearly 0. That is, SocialTrust can effectively combat collusion nodes. By considering social closeness and the interest relationship between nodes, SocialTrust reduces the impacts of the ratings from the potential colluders and reduces the reputation values of the colluders.

5.4 Pair-wise collusion (PCM) with compromised pretrusted nodes

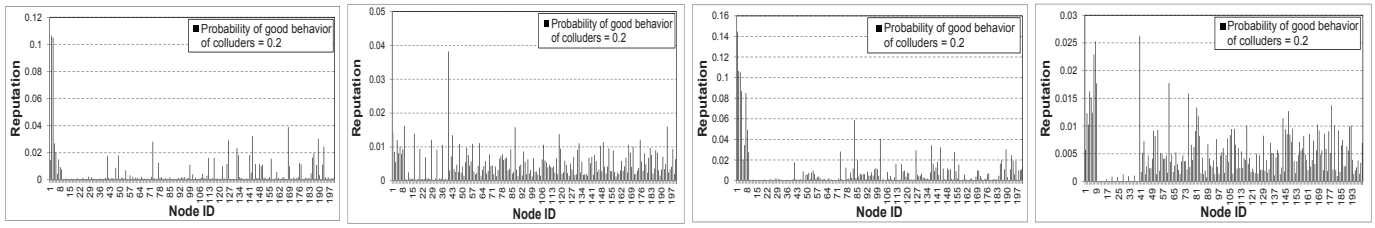
We consider a scenario where B=0.2 and compromised pretrusted nodes are involved in the collusion. We randomly selected 7 nodes from the pretrusted nodes and let them randomly select a colluder with which to collude. We set the social distance between a compromised pretrusted node and its conspired colluder to 1.

Figure 10(a) shows the reputation distribution of the nodes in EigenTrust. Comparing Figure 10(a) with Figure 9(a), we find that compromised pretrusted nodes greatly boost the reputations of themselves and colluders, and they reduce the reputations of normal nodes



(a) EigenTrust. (b) eBay. (c) EigenTrust with SocialTrust. (d) eBay with SocialTrust.

Fig. 11: Reputation distribution in MCM with B=0.6 (pretrusted nodes: 1-9, colluders: 10-39).



(a) EigenTrust. (b) eBay. (c) EigenTrust with SocialTrust. (d) eBay with SocialTrust.

Fig. 12: Reputation distribution in MCM with B=0.2 (pretrusted nodes: 1-9, colluders: 10-39).

accordingly. This is due to three reasons. First, the ratings of pretrusted nodes have higher weight and they rate highly on the colluders, causing the reputations of the colluders in collusion with the pretrusted nodes to increase. Second, because of the high reputations of these colluders, their ratings for the pretrusted nodes also have higher weight, further boosting pretrusted nodes' already high reputations. Third, as the colluders mutually rate each other with high frequency, the reputations of all colluders are boosted. The result implies that malicious nodes can take advantage of EigenTrust's pretrusted node strategy by compromising these nodes, which helps them to quickly boost their own reputations. EigenTrust cannot deal with the challenge of collusion involvement of compromised pretrusted nodes.

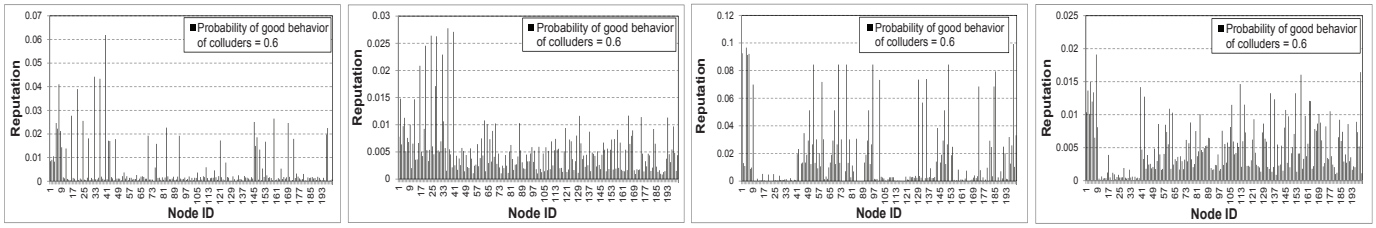
Figure 10(b) shows the reputation distribution of the nodes in EigenTrust with SocialTrust in the same scenario. We observe that high-reputed nodes are skewed among normal nodes and the non-compromised pretrusted nodes. The reputations of the colluders and pretrusted nodes involved in collusion have nearly 0 reputations. The pretrusted nodes have high probability to provide authentic services and receive high reputations accordingly. SocialTrust detects the pairs of suspicious colluders, including the compromised pretrusted nodes, which have a high mutual rating frequency. It then adjusts their reputations according to their social closeness and interest similarity. Therefore, although a compromised pretrusted node initially has a high reputation, its reputation eventually drops to a low value. The results demonstrate the capability of SocialTrust in countering collusion even when pretrusted nodes are compromised.

5.5 Multiple node collusion (MCM)

In the multiple node collusion model, among the 30 colluders, 7 nodes are randomly selected as the boosted nodes, and all other colluders randomly select one of the boosted nodes to collude with. Figure 11(a) shows the reputation distribution of nodes in EigenTrust when B=0.6. It demonstrates that some colluders, which are boosted nodes, have very high reputations while other colluders, which are boosting nodes, have very low reputations. This is caused by two reasons. First, as the colluders offer authentic services to others with probability of 0.6, they can initially gain reputations. Second, since the boosted nodes frequently receive positive ratings from several boosting nodes whose reputation values are not low, the reasonable rating weight of the boosting nodes can greatly increase the reputation value of the boosted nodes. The boosting nodes do not receive frequent ratings from the boosted nodes. As the boosted nodes receive more and more service requests, the boosting nodes receive fewer and fewer requests, causing fewer opportunities to raise their reputations.

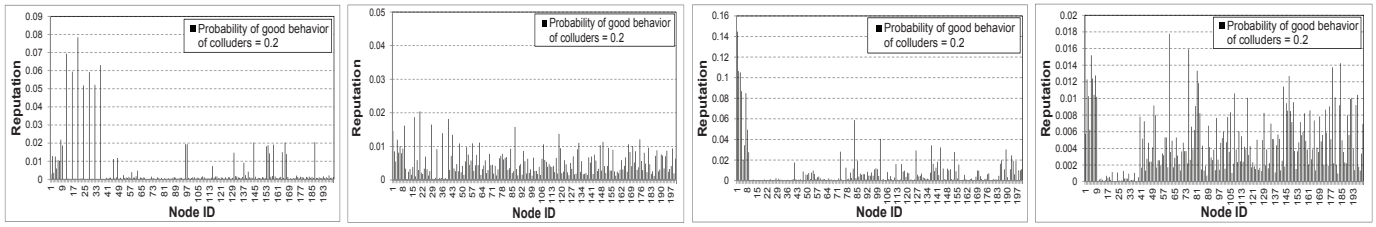
Figure 11(b) plots the reputation distribution of the nodes in eBay. It shows that the reputation values of some of the colluders are much higher than other nodes in the system while other colluders have comparatively lower reputations. This is due to the same reason in Figure 11(a). Figure 11(c) plots the reputation distribution of nodes in EigenTrust with SocialTrust. By comparing it to Figure 11(a), we see that SocialTrust can effectively reduce the reputation values of both boosted and boosting nodes in EigenTrust. Although boosted nodes can receive a large number of positive ratings from boosting nodes, the values of these ratings are reduced according to the social and interest relationship between the raters and ratees. Meanwhile, due to the low reputation values of those boosting nodes, the weights of their ratings are very low. Therefore, it is difficult for them to increase the reputation values of boosted nodes even with high rating frequency. Figure 11(d) shows the reputation distribution of the nodes in eBay with SocialTrust. It shows that SocialTrust can effectively fight against collusion. Although the boosting nodes can increase the reputation values of the boosted nodes as shown in Figure 11(b), SocialTrust reduces the impact of the rating between the colluders based on their social closeness and interest similarity, and the reputation values of the colluders are reduced significantly in SocialTrust.

Next, we changed the probability that the colluders provide authentic services to 0.2. Figure 12(a) shows the reputation distributions of the nodes in EigenTrust. It shows that the reputations of the colluders including



(a) EigenTrust. (b) eBay. (c) EigenTrust with SocialTrust. (d) eBay with SocialTrust.

Fig. 13: Reputation distribution in MMM with B=0.6 (pretrusted nodes: 1-9, colluders: 10-39).



(a) EigenTrust. (b) eBay. (c) EigenTrust with SocialTrust. (d) eBay with SocialTrust.

Fig. 14: Reputation distribution in MMM with B=0.2 (pretrusted nodes: 1-9, colluders: 10-39).

the boosted nodes are very low. Two factors contribute to this phenomenon. First, as the boosting nodes have low reputation values, the weight of their ratings is small. Therefore, their frequent ratings cannot affect the reputations of the boosted nodes. Second, as the boosted nodes have a high probability to provide inauthentic service, the ratings they receive from other normal nodes are very low. EigenTrust can counter MCM when the colluders provide authentic services with low probability.

Figure 12(b) shows the reputation distribution of nodes in eBay. We can see that the reputation values of some colluders are low while others are comparatively high. Since the probability that the colluders offer authentic services is only 0.2, they receive low reputation values from normal nodes. The boosted nodes receive a large number of positive ratings from boosting nodes. Since the rating values from low reputed boosting nodes are not weighted, they can partially offset the negative ratings from normal nodes. Consequently, the reputation values of the boosted nodes are increased incrementally. Figures 12(c) and (d) show the reputation distribution of the nodes in EigenTrust and eBay with SocialTrust, respectively. The figures show that SocialTrust further reduces the reputation values of the boosted nodes. The results demonstrate the effectiveness of SocialTrust in combating collusion by considering their social and interest relationships.

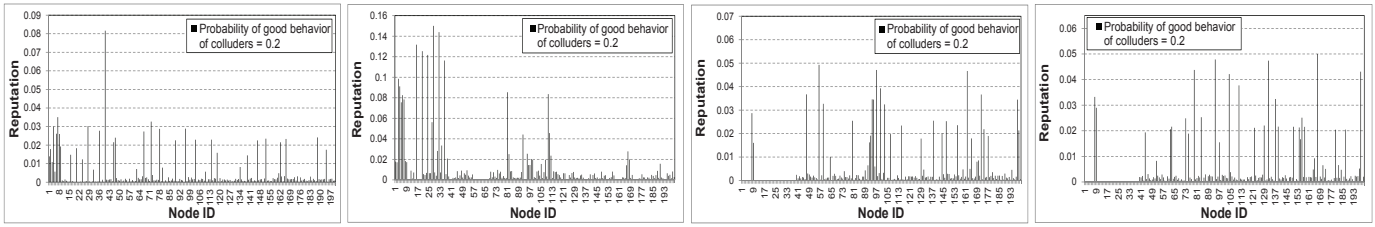
5.6 Multiple and mutual node collusion (MMM)

In this section, we evaluate the effectiveness of EigenTrust, eBay and SocialTrust in combating collusion in MMM. In every query cycle, each boosting node rates randomly chosen boosted nodes 20 times and the boosted node rates its boosting nodes 5 times. Figures 13(a) and (b) show the reputation distribution of the nodes in EigenTrust and eBay, respectively. Comparing these two figures to Figure 11(a) and (b), we see that in MMM, the reputation values of both boosted nodes and boosting nodes are much higher. Since the colluders offer authentic services with probability 0.6, they gain a certain amount of reputations. In MMM, as the boosted nodes with high reputations rate the boosting nodes back, the reputation values of boosting nodes are also enhanced. Then, the weights of boosting nodes' ratings increase, which in return boosts the reputation values of boosted nodes. Figures 13(c) and (d) illustrate the node reputation distribution of nodes in EigenTrust and eBay with SocialTrust, respectively. The figures show that the reputation values of colluders are greatly reduced due to the same reasons as Figures 11(c) and (d).

Figure 14(a) shows the reputation distribution of the nodes in EigenTrust in MMM with B=0.2. Comparing Figure 14(a) with Figure 12(a) for MCM, we find that in Figure 14(a), some colluders have extremely high reputation, which are even higher than those of the pretrusted nodes. These colluders are the boosted nodes. In MMM, the boosted nodes and boosting nodes mutually rate each other. Therefore, the reputation values of the boosting nodes are initially low, and they are increased as the boosted nodes having high rating weights rate them back. The boosting nodes then in turn greatly increase the reputation values of the boosted nodes. Although boosted nodes rate boosting nodes at a lower frequency, the boosted nodes still reach very reputations.

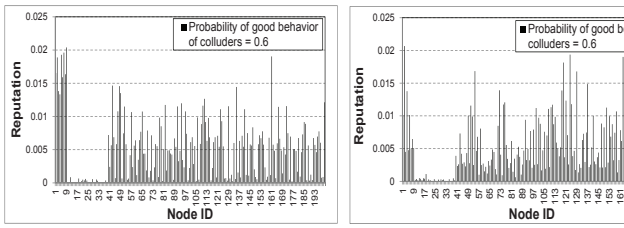
In PCM with B=0.2 (Figure 9(a)), the colluders cannot boost their reputations because their rating frequency of 20 per query cycle is not high enough to offset the low reputations from normal nodes. In contrast, in MMM, a boosted node receives 80 ratings per query cycle on average from boosting nodes. Therefore, although they receive low ratings from normal nodes, their reputations can still be increased. The results demonstrate that EigenTrust is not effective in countering MMM where nodes rate each other with a high frequency.

Comparing Figure 14 and Figure 13, the reputation values of colluders in EigenTrust with B=0.2 are even higher than those with B=0.6. Those colluders are actually boosted nodes. With B=0.6, as the boosting nodes gain some reputations, they receive certain service requests from normal nodes. Therefore, the boosted node receives less service requests and lower reputation values. When B=0.2, boosting nodes have very low reputation and subsequently cannot receive many service requests from the normal nodes. Consequently, the boosted nodes receive more service requests and



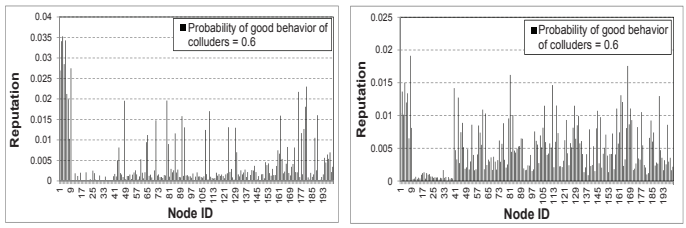
(a) EigenTrust in MCM. (b) EigenTrust in MMM. (c) EigenTrust with SocialTrust in MCM. (d) EigenTrust with SocialTrust in MMM.

Fig. 15: Reputation distribution in MCM and MMM with compromised pretrusted node with B=0.2 (pretrusted nodes: 1-9, colluders: 10-39).



(a) EigenTrust with SocialTrust. (b) eBay with SocialTrust.

Fig. 16: Reputation distribution in PCM with B=0.6 (pretrusted nodes: 1-9, colluders: 10-39).



(a) EigenTrust with SocialTrust. (b) eBay with SocialTrust.

Fig. 17: Reputation distribution in MCM with B=0.6 (pretrusted nodes: 1-9, colluders: 10-39).

comparatively higher reputations.

Figure 14(b) shows the reputation distribution of the nodes in eBay. Comparing this figure with Figure 12(b) for MCM, we find that the reputation values of boosting nodes in MMM are slightly higher. The reason is that the boosting nodes also receive ratings from boosted nodes, which increases the reputation values of the boosting node. Figures 14(c) and (d) show the reputation distribution of the nodes in EigenTrust and eBay with SocialTrust, respectively. Comparing these two figures to Figures 14(a) and (b), we see that SocialTrust can effectively reduce the reputation of the colluders due to the same reasons as Figures 11(c) and (d).

5.7 Node collusion (MCM and MMM) with compromised pretrusted nodes

Figure 15(a) and (b) demonstrate the reputation distribution of the nodes in EigenTrust in MCM and MMM, respectively, when compromised pretrusted nodes are involved in collusion with B=0.2. Colluders and pretrusted nodes collude in the same way as Figure 10. Comparing Figure 15(a) to Figure 12(a) for MCM, we see that when pretrusted nodes are involved in collusion, the reputations of some colluders increase greatly while those of pretrusted nodes decrease. Because of B=0.2, boosting nodes have low reputations and a low weight for their ratings. As shown in Figure 12(a), their frequent ratings on the boosted nodes cannot greatly increase their reputations. The reputation values of the pretrusted nodes are high. Therefore, when pretrusted nodes are compromised, as shown in Figure 15(a), their ratings greatly increase the reputations of the boosted nodes, which attract many requests from the pretrusted nodes.

Figure 15(b) shows the reputation distribution of the nodes in EigenTrust in MMM. It shows that the reputation values of colluders are 3-4 times higher than those of the colluders when pretrusted nodes are not involved in collusion (Figure 14(a)). Because the pretrusted nodes give positive ratings on colluders with high frequency, the reputation values of the boosted nodes become very large, and the weight of their ratings increases accordingly. This greatly increases the reputations of colluders through frequent ratings between each other. Figures 15(c) and (d) show the reputation distribution of the nodes in EigenTrust with SocialTrust in MCM and MMM, respectively. The figures show that both the colluders and compromised pretrusted nodes have low reputations. It means that SocialTrust can still effectively reduce the reputation values of the colluders and compromised pretrusted nodes based on the social and interest relationship between the nodes.

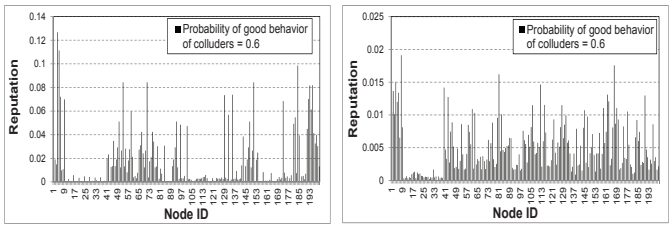
5.8 Resilience to falsified social information

In this experiment, we assume colluders falsify their social information so that each pair of colluders has only one social relationship and identical interests. The number of identical interests is randomly chosen from [1-10]. The frequency of requests on a node's interest $w_{c_{(i,l)}}$ equals the percent of requests on this interest in all the node's requests in a simulation cycle. Previous experimental results show that when B=0.6, both EigenTrust and eBay with SocialTrust can effectively detect the colluders. We now test their resilience to falsified social relationships and interests.

Figure 16, Figure 17, and Figure 18 show the reputation distribution of both EigenTrust and eBay with SocialTrust with falsified social information in PCM, MCM and MMM. By comparing the results in these figures with the results with accurate social information in Figure 8(c) and (d), Figure 11(c) and (d), and Figure 13(c) and (d) for each model, we find that although the reputation values of colluders are higher with falsified social information than those with accurate social information, their reputation values are still significantly lower than those of other nodes. It implies that SocialTrust still can effectively thwart collusion even when the social information is falsified. In addition to the social information in user profiles, SocialTrust also considers user interaction frequency and resource request frequency which cannot be falsified and can truly reflect real user social closeness and social interest.

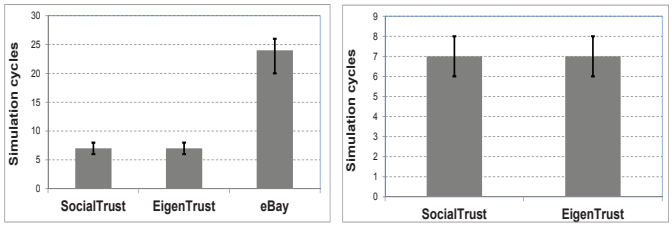
5.9 Efficiency and effectiveness in combating colluders

We measured the number of simulation cycles until the reputation values of colluders are lower than 0.001. Figure 19(a) shows the 1st percentile, 99th percentile and median values of the number of simulation cycles in EigenTrust, SocialTrust and eBay when B=0.2. We see that both EigenTrust and SocialTrust take about 6-8 simulation cycles, while eBay takes about 25 simulation cycles. This is because the reputation calculation methods in EigenTrust and SocialTrust make the node reputation values converge very fast, while the reputation calculation in eBay is not sufficiently efficient. Figure 19(b) shows the number of simulation cycles in EigenTrust and SocialTrust when B=0.6. It exhibits similar performance as in Figure 19(a) due to their fast reputation convergence. We did not plot eBay in this figure because eBay cannot detect colluders when B=0.6. We also measured the number of simulation cycles of Eigentrust, SocialTrust and eBay in PCM and MCM. The experimental results are almost the same as those in MMM, which confirms the high efficiency of collusion deterrence in EigenTrust and SocialTrust. Due to the space limit, we do not show these experimental here.



(a) EigenTrust with SocialTrust. (b) eBay with SocialTrust.

Fig. 18: Reputation distribution in MMM with B=0.6 (pretrusted nodes: 1-9, colluders: 10-39).



(a) B=0.2 (b) B=0.6

Fig. 19: Efficiency for combating colluders in MMM.

Figure 20 shows the average reputation values of the colluders versus different social distances between colluders in PCM, MMM and MCM. We see that the average reputation of colluders increases slightly when the social distance increases from 1 to 2, but decreases slightly when it increases to 3 subsequently. Recall that the distance between a pair of nodes was randomly chosen from [1,3]. Social distance 1 means social closeness is high while social distance 3 means social closeness is low. As shown in Equation (9), the rating between two nodes with too close or too far-away social relationship is reduced more. We also see that the reputation values of the colluders are constantly lower than those of normal nodes even when the colluders have moderate social distance 2. This result confirms the effectiveness of SocialTrust in mitigating the adverse influence of collusion even when colluders keep their social distance at a normal level. SocialTrust reduces the rating value based on both social closeness and interest similarity, which are very difficult to be manipulated as explained in Section 4.4.

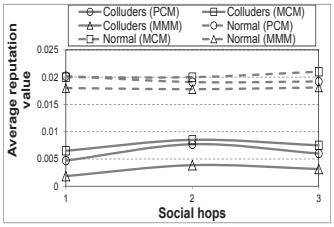


Fig. 20: Average reputation vs. social distance.

5.10 Percentage of requests sent to colluders

Table 1 shows the percentage of requests sent to colluders in each system in different collusion models with B=0.2 and B=0.6, respectively. In the table, "(Pre)" means that the pretrusted nodes are involved in collusion. First, we see that in all three collusion models, colluders receive more requests when B=0.6 than when B=0.2 in most systems. This is because colluders with higher probability to provide authentic services have higher reputation values initially, which leads to higher weight for their ratings and hence further enhance their reputations, finally attracting more requests from the normal nodes. Second, comparing the results in different collusion models, we find that more service requests are sent to colluders in MMM and PCM than MCM. This is because colluders in MMM and PCM mutually rate each other with high frequency while boosting nodes in MCM do not receive ratings from boosted nodes.

TABLE 1: Percentage of the requests sent to colluders.

Pair-wise collusion model (PCM)			
B=0.2		B=0.6	
eBay	6%	eBay	17%
EigenTrust	17%	EigenTrust	24%
EigenTrust (Pre)	22%	EigenTrust (Pre)	24%
eBay+SocialTrust	3%	eBay-Social	2%
EigenTrust+SocialTrust	2%	EigenTrust+SocialTrust	3%
EigenTrust+SocialTrust (Pre)	2%	EigenTrust+SocialTrust (Pre)	2%
Multiple node collusion model (MCM)			
B=0.2		B=0.6	
eBay	7%	eBay	16%
EigenTrust	7%	EigenTrust	15%
EigenTrust (Pre)	9%	EigenTrust (Pre)	10%
eBay+SocialTrust	3%	eBay+SocialTrust	2%
EigenTrust+SocialTrust	2%	EigenTrust+SocialTrust	2%
EigenTrust+SocialTrust (Pre)	2%	EigenTrust+SocialTrust (Pre)	2%
Multiple and mutual node collusion model (MMM)			
B=0.2		B=0.6	
eBay	8%	eBay	17%
EigenTrust	19%	EigenTrust	21%
EigenTrust (Pre)	21%	EigenTrust (Pre)	24%
eBay+SocialTrust	2%	eBay+SocialTrust	2%
EigenTrust+SocialTrust	3%	EigenTrust+SocialTrust	3%
EigenTrust+SocialTrust (Pre)	4%	EigenTrust+SocialTrust (Pre)	3%

Third, in EigenTrust and eBay in all collusion models, the percent of requests sent to colluders when pretrusted nodes are involved in collusion is higher than when they are not involved in collusion in most cases. This is because the pretrusted nodes increase the reputation values of colluders, which subsequently attract more service requests. Finally, we see that SocialTrust reduces the percent of requests sent to colluders to $2\% - 4\%$ in different systems and collusion models, even when pretrusted nodes are involved in the collusion. By considering the social closeness and interest similarity, SocialTrust adjusts the ratings between the suspected colluders. Then, these nodes receive low reputations and fewer service requests, which discourages the collusion behaviors.

6 Conclusion

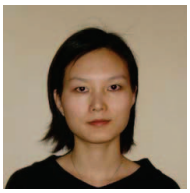
Despite the effectiveness of reputation systems in finding deceptive peers according to the reputation values, they

are vulnerable to collusion. Although many reputation systems try to reduce the influence of collusion on reputation values, they are not sufficiently effective in countering collusion. After examining the Overstock transaction trace of reputation ratings, we identified suspicious collusion behavior patterns. According to the behavior patterns, we propose the SocicalTrust mechanism that leverages a social network to combat collusion. Experiment results show SocicalTrust greatly enhances the capability of eBay's reputation system and EigenTrust in countering collusion. SocicalTrust can even detect colluders with compromised pretrusted high-reputed nodes and falsified social information. In our future work, we will further investigate other collusion patterns, security issues and attack models in the SocialTrust design.

REFERENCES

[1] Bittorrent. www.bittorrent.com/.
[2] PPLive. http://www.pplive.com.
[3] M. Cai, M. Frank, J. Chen, and P. Szekely. MAAN: a multi-attribute addressable network for grid information services. *Journal of Grid Computing*, 2004.
[4] eBay. http://www.ebay.com.
[5] Amazon. http://www.amazon.com/.
[6] Overstock. http://www.overstock.com/.
[7] Q. Lian, Z. Zhang, M. Yang, B. Y. Zhao, Y. Dai, and X. Li. An empirical study of collusion behavior in the MAZE P2P file-sharing system. In *Proc. ICDCS*, 2007.
[8] S. Zhao and V. Lo. Result verification and trust-based scheduling in open peer-to-peer cycle sharing systems. In *Proc. of P2P*, 2005.
[9] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In *Proc. of WWW*, 2003.
[10] M. Schlosser S. Kamvar and H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In *Proc. of WWW*, 2003.
[11] M. Yang, Y. Dai, and X. Li. Bring reputation system to social network in the Maze P2P file-sharing system. In *Proc. of CTS*, 2006.
[12] M. Srivatsa, L. Xiong, and L. Liu. Trustguard: Countering vulnerabilities in reputation management for decentralized overlay networks. In *Proc. of WWW*, 2005.
[13] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust incentive techniques for peer-to-peer networks. In *Proc. of EC*, 2004.
[14] Z. Liang and W. Shi. Pet: A personalized trust model with reputation and risk evaluation for P2P resource sharing. In *Proc. of HICSS*, 2005.
[15] R. Sherwood S. Lee and B. Bhattacharjee. Cooperative peer groups in NICE. In *Proc. of INFOCOM*, 2003.
[16] E. Zhai, R. Chen, Z. Cai, L. Zhang, E. K. Lua, H. Sun, S. Qing, L. Tang, and Z. Chen. Sorcery: Could we make P2P content sharing systems robust to deceivers? In *Proc. of P2P*, 2009.
[17] T. Isdal, M. Piatek, A. Krishnamurthy, and T. Anderson. Privacy-preserving P2P data sharing with OneSwarm. In *Proc. of SIGCOMM*, 2010.
[18] L. Xiong and L. Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *TKDE*, 2004.
[19] A. Singh and L. Liu. TrustMe: Anonymous management for trust relationships in decentralized P2P systems. In *Proc. of P2P*, 2003.
[20] R. Zhou and K. Hwang. Powertrust: A robust and scalable reputation system for trusted P2P computing. *TPDS*, 2007.
[21] S. Song, K. Hwang, R. Zhou, and Y.-K. Kwok. Trusted P2P transactions with fuzzy reputation aggregation. *IEEE Internet Computing*, 2005.
[22] R. Zhou and K. Hwang. Gossip-based reputation management for unstructured peer-to-peer networks. *IEEE TKDE*, 2007.
[23] C. P. Costa and J. M. Almeida. Reputation systems for fighting pollution in p2p file sharing systems. In *Proc. of P2P*, 2007.
[24] K. Walsh and E. Sirer. Experience with an object reputation system for peer-to-peer file sharing. In *Proc. of NSDI*, 2006.
[25] F. Cornelli, E. Damiani, S. Vimercati, S. Paraboschi, and P. Samarati. Choosing Reputable Servents in a P2P Network. In *Proc. of WWW*, 2002.
[26] E. Damiani, S. D. C. Vimercati, S. Paraboschi, P. Samarati, and F. Violante. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *Proc. of CCS*, 2002.
[27] N. Curtis, R. Safavi-Naini, and W. Susilo. X^2Rep: Enhanced Trust Semantics for the XRep Protocol. In *Proc. of ACNS*, 2004.
[28] T. Moreton and A. Twigg. Trading in trust, tokens, and stamps. In *Proc. of P2PEcon*, 2003.
[29] H. Yu, M. Kaminsky, and A. Flaxman. SybilGuard: Defending against Sybil attacks via social networks. In *Proc. of Sigcomm*, 2006.
[30] H. Yu, P. B. Gibbons, and M. Kaminsky. SybilLimit: A near-optimal social network defense against Sybil attacks. In *Proc. of S&P*, 2008.
[31] George Danezis and Prateek Mittal. SybilInfer: Detecting Sybil nodes using social networks. In *Proc. of NDSS*, 2009.
[32] D. Nguyen, B. Min, J. Li, and L. Subramanian. Sybil-resillient online content rating. In *Proc. of NSDI*, 2009.
[33] B. Viswanath and A. Post. An analysis of social network-based sybil defenses. In *Proc. of SIGCOMM*, 2010.
[34] C. Lesniewski-Laas. A Sybil-proof one-hop DHT. In *Proc. of SNS*, 2008.
[35] C. Lesniewski-Laas and M. Frans Kaashoek. A Sybil-proof distributed hash table. In *Proc. of NSDI*, 2010.
[36] A. Fast, D. Jensen, and B. N. Levine. Creating social networks to improve peer to peer networking. In *Proc. of KDD*, 2005.
[37] R. Xiang, J. Neville, and M. Rogati. Modeling relationship strength in online social networks. In *Proc. of WWW*, 2010.
[38] J. Leskovec, D. Hutener, and J. Kleinberg. Predicting positive and negative links in online social networks. In *Proc. of WWW*, 2010.
[39] C. Binzel and D. Fehr. How social distance affects trust and cooperation. In *Proc. of ERF*, 2009.
[40] G. Swamynathan, C. Wilson, B. Boe, K. Almeroth, and B. Y. Zhao. Do social networks improve e-commerce? a study on social marketplaces. In *Proc. of WOSN*, 2008.
[41] S. Garriss, M. Kaminsky, M. J. Freedman, B. Karp, D. Mazieres, and H. Yu. Re: Reliable email. In *Proc. of NSDL*, 2006.
[42] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 2001.
[43] Z. Li and H. Shen. Soap: A social network aided personalized and effective spam filter to clean your e-mail box. In *Proc. of Infocom*, 2011.
[44] A. Iamnitchi, M. Ripeanu, and I. Foster. Small-world file-sharing communities. In *Proc. of INFOCOM*, 2004.

Ze Li Ze Li received the BS degree in Electronics and Information Engineering from Huazhong University of Science and Technology, China, in 2007. He is currently a Ph.D. student in the Department of Electrical and Computer Engineering of Clemson University. His research interests include distributed networks, with an emphasis on peer-to-peer and content delivery networks. He is a student member of IEEE.



Haiying Shen Haiying Shen received the BS degree in Computer Science and Engineering from Tongji University, China in 2000, and the MS and Ph.D. degrees in Computer Engineering from Wayne State University in 2004 and 2006, respectively. She is currently an Assistant Professor in the Department of Electrical and Computer Engineering at Clemson University. Her research interests include distributed computer systems and computer networks, with an emphasis on P2P and content delivery networks, mobile computing, wireless sensor networks, and grid and cloud computing. She was the Program Co-Chair for a number of international conferences and member of the Program Committees of many leading conferences. She is a Microsoft Faculty Fellow of 2010 and a member of the IEEE and ACM.