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

Ze Li, Haiying Shen and Karan Sapra
Department of Electrical and Computer Engineering
Clemson University, Clemson, SC, 29631
Email: {zel, shenh, ksapra}@clemson.edu

***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 important 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, namely SocialTrust, to counter collusion. SocialTrust adaptively adjusts the weight of ratings based on the social distance and interest relationship between peers. Experimental results show that SocialTrust can significantly strengthen the capability of current reputation systems in combating collusion.**

***Keywords*-Social networks; P2P networks; Collusion detection;**

I. Introduction

The past decade has seen a rapid development of peer-to-peer networks (P2Ps) along with a dramatic surge of real or potential applications including file sharing (e.g., BitTorrent [1] and Gnutella [2]), video streaming sharing (e.g., PPLive [3]), computing resource sharing (e.g., MAAN [4]). In all of these P2P applications, peers (acquaintance and non-acquaintance) directly contact with each other to conduct transactions on resources (e.g., files, videos and computing resources).

Considering P2Ps' open environment where many autonomous nodes without preexisting trust relationships often share resources or conduct transactions with each other, a critical problem is how can a resource requester choose a resource provider that is trustworthy and provides high-quality service (QoS) among many resource providers?

To deal with this problem, P2Ps usually employ reputation systems for reliability and security. Like the reputation systems in the eBay [5], Amazon [6] and Overstock [7] online auction platforms, a reputation system employed in P2Ps computes and publishes global reputation value for each node based on a collection of local ratings from others about the node in order to provide guidance in selecting trustworthy nodes. However, reputation systems are generally vulnerable to node collusion [8, 9], which impairs their effectiveness in trustworthy server 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 [10].

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. The reputation system in eBay [11] counts only one positive or negative rating for multiple ratings from one user to another in one week. This means even though a peer conducts several transactions with another peer in one week, only one rating is considered, which affects the accuracy of global reputation values. Some mechanisms assign a higher weight to ratings from pretrusted peers and (or) assigns weights to ratings according to the raters' global reputations [10, 12, 13]. 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 [14–17] of itself or its friends [18–20]. However, these mechanisms limit the server options and prevent non-acquaintance from freely conducting transactions between each other.

In this paper, we propose a mechanism called SocialTrust that leverages social networks to enhance the effectiveness of current mechanisms in combating collusion. A social network is a social structure consists of individuals (nodes) that are tied by one or more specific types of relationship, such as common interests, friendship, kinship or trade [21].

To investigate the impact of a social network on user purchasing and rating patterns, we analyzed a real trace of 450,000 transaction ratings during 2008-2010 that we crawled from Overstock Auctions (Overstock in short) [7]. 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. Also, users tend to buy products from socially-close (3 hops or less) users, and rate socially-close users with high ratings. Second, 88% of the purchases of a user is within 20% of the user's product interest categories 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 peer reputations. Based on our observations, we identified suspicious collusion behavior patterns based on the distance and interest relationship between peers in a social network. The ratings from suspected colluders include: (1) frequent high ratings between low-reputed peers with short social distance, since peers seldom request resources from low-reputed peers. (2) frequent high ratings from nodes with long social distance, since peers tend to request resources from socially-close peers. (3) frequent high ratings between nodes with low interest similarity, since peers request resources in their interests most of the time. (4) frequent low ratings from nodes with high interest similarity, since such nodes may be competitors in attracting requests for similar resources. SocialTrust adjusts these ratings according to node social closeness and interest similarity in order to reduce the impact of collusion on reputations.

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 following three contributions.

(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 collusion. SocialTrust adjusts the ratings from suspected colluders based on social closeness and interest similarity between a rater and a ratee.

(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.

II. Related Work

Many reputation systems have been proposed which assign reputation values based on performance measures of peers, and then find deceptive peers according to the reputation values. These systems include PeerTrust [22], Trustme [23], EigenTrust [24], PowerTrust [25], TrustGuard [13], FuzzyTrust [26], GossipTrust [27], and Scrubber [28]. PeerTrust [22] computes peer reputation scores based on three basic trust parameters and two adaptive factors. The three parameters include the feedback a peer receives from other peers, the total number of transactions a peer performs, and the credibility of the feedback sources. The two adaptive factors include transaction context factor and the community context factor. Trustme [23] offers an approach toward anonymous trust management which can provide mutual anonymity for both the trust host and the trust querying peer. EigenTrust [24] and PowerTrust [25] depend on the P2P reputation exchange to calculate the global reputation value of each peer based on the distributed hash table. TrustGuard [13] incorporates historical reputations and behavioral fluctuations of nodes into the estimation of their trustworthiness. It improves system robustness by guaranteeing that reputation is built gradually, but drops quickly when a node starts to behave maliciously. FuzzyTrust [26] uses fuzzy logic inferences, which can better handle uncertainty, fuzziness, and incomplete information in peer trust reports. In GossipTrust [27], peers sharing weighted local trust scores with randomly selected neighbors until reaching global consensus on peer reputations. Costa *et al.* [28] proposed to use a reputation system to fight polluted file content by rating both the file provider and file.

Credence [29] is designed to give users robust estimate of file authenticity, which means the degree to which an object's content matches its advertised description. Cornelli *et al.* [30] proposed an approach to P2P security where servants can keep track, and share with others information about the reputation of their peers. It enables each client to compute a personalized, rather than global, performance score for peers, and also distinguish peer performance from peer credibility. Both XRep [19] and X^2Rep [20] extend the work in [30] by additionally computing object reputations based on weighted peer voting.

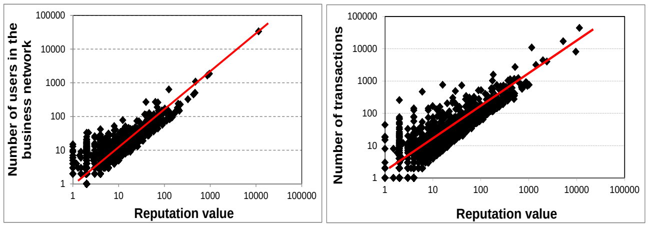
Recently, a number of research works have been conducted on the problem of collusion in reputation systems. EigenTrust [24] breaks collusion collectives by assigning a higher weight to the feedback of pretrusted peers. Yang *et al.* [12] introduced using social networks in the Maze P2P file sharing system 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. In Sorcery [18], each client establishes its confidential and reliable friend-relationship social network. Clients utilize the overlapping voting histories of both their friends and the content providers, and judge whether a content provider is a colluder. Moreton *et al.* proposed the Stamp algorithm [31], 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. The Stamp algorithm captures the essence of both reputation and payment protocols, and can inhibit collusion behaviors. Srivatsa *et al.* proposed the notion of personalized credibility measurement in which the feedbacks from similar raters are given a higher weight [13]. It acts as an effective defense against potential collusive nodes that only give good ratings within the clique and give bad rating to the clique outside. Lian *et al.* [8] analyzed the traffic logs in a P2P file sharing system to study different types of collusion patterns.

All previous methods that use social networks to handle collusion are based on the rationale that the ratings from friends are trustable. However, these methods limit the server options and constrain resource sharing to only between friends. They also cannot provide a global reputation of each node calculated by ratings from a variety of users to accurately reflects 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.

III. Analysis of Real Trace 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. Similar to eBay's reputation system, 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 (social) network" and a "business network." The "personal network" is a social network that comprises of friends invited by the user. In the personal page of the personal network, a user can list hobbies and interests, post photos, and publish friends and business contact lists. 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.

In order to study the relationship between 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. We identified suspicious collusion behavior patterns based on two main characteristics of collusion described in [8, 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.



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

(b) # of transactions vs. reputation of each user

Figure 1: Effect of reputation on transaction.

A. Relationship between reputation, social network and transaction

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, users 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 in proportional to his/her reputation. It 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:

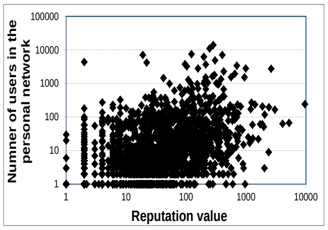


Figure 2: Social network size vs. reputation (C=0.092)

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.

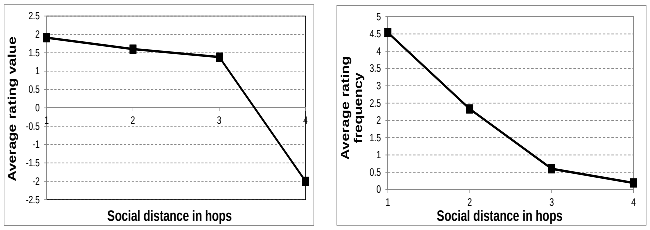
Figure 2 shows the number of users in the personal network of each user versus her/his reputation value. We can see that there is a very weak linear relationship between personal network size and reputation value. Their correlation coefficient is only 0.092. The linear relationship may be caused by the 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.

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.

B. Impact of social closeness

Social distance between two users in the social network graph represents the social closeness between the two users. If two users are directly connected in the personal network, their social distance is 1. If one user is a friend of another user's friend, then the social distance between them is 2, and so on. Next, we investigate the impact of social distance on user purchasing and reputation rating behavior.



(a) Ave. value of ratings of a rater per day.

(b) Ave. # of ratings of a rater per day.

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

Our crawled data shows that there are no transactions between users with > 4 hop social distance. 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 can see that as the social distance between people increases, the average rating values and average number of ratings decrease.

O3: All transactions occur between users with short social distances (4 hops or less) and most transactions occur between users within 3 hops.

Thus, we identify a suspicious behavior of collusion:

Suspicious behavior 1 (B1): Users with long social distances rate each other with high ratings and high frequency.

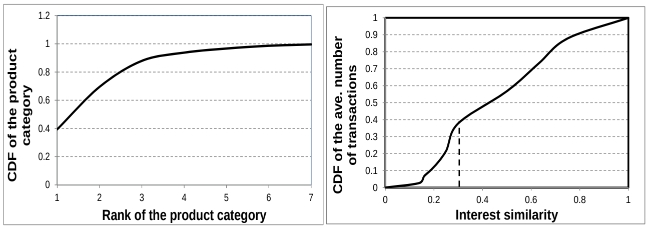
O4: Users with shorter social distances are more likely to rate each other with higher ratings and higher frequency.

From I1, I2 and O4, we get:

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

C. Impact of social interest similarity

Next, we investigate the impact of user interest on user purchasing pattern. We classified the products bought or sold by the users into categories such as Electronics, Computers, and Clothing. We then generated an interest vector $\mathcal{V}$=$<v_1, v_2, v_3, ..., v_k>$ for each user, where v denotes a product category. We ranked the categories of the products that each buyer has purchased 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,



(a) CDF of the top 7 category ranks.

(b) Ave. number of transactions vs. interest similarity between a pair .

Figure 4: Impact of interests on purchasing pattern.

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

It was indicated that normal nodes primarily request items in their interests [32]. 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

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

Figure 4(b) depicts the CDF of the average number of transactions versus interest similarity. We see only 10% 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 the competitors with low ratings. Thus, we identify another suspicious behavior:

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

IV. SocialTrust: Social Network based Mechanism to Combat Collusion

Based on the suspicious collusion behaviors observed in Section III, we propose a social network based mechanism

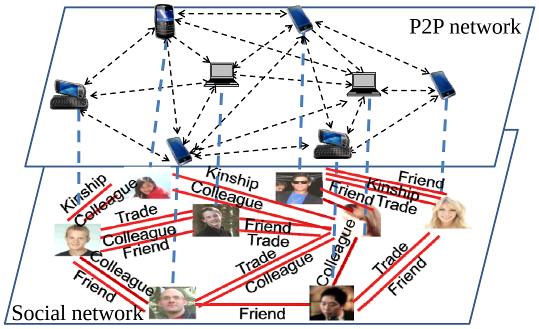


Figure 5: A social network graph.

to combat collusion, called SocialTrust. SocialTrust can be used on any reputation system to enhance its capacity to combat collusion. If a P2P network already incorporates an online social network like Overstock, SocialTrust can directly uses the social network. Otherwise, SocialTrust provides a plugin for the social network construction. Specifically, it requires users to input their interest information. It also establishes friend-relationship (acquaintances in reality or reliable online friends) social network as in other reputation systems [12, 18, 33].

As shown in Figure 5, a social network provides a graphic view of the interdependency of the subjective human relationship in our society, such as personal relationship and interest activity communities. SocialTrust derives the *social closeness* (from the social network graph and node interaction) and *interest similarity* (from node profiles or activities) between a pair of nodes. We use Ω_d and Ω_c 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.

Social closeness: We first introduce a method to calculate the social closeness between two adjacent nodes in a social network, and then introduce a method for non-adjacent nodes having no direct social relationship. The closeness of a pair of nodes n_i and n_j is determined by two factors: the number of social relationships and interaction frequency. More relationships between two nodes means 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. The social network provides social relationship information such as colleague and classmate. For social interaction information, we regard the action that a peer requests a resource from another peer in the P2P and (or) a peer posts a comment on another peer's wall in the social network as an interaction. Therefore, considering the two factors, the social closeness $\Omega_{d_{(i,j)}}$ between two adjacent nodes n_i and n_j is calculated by

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

where $m_{(i,j)}{\geq}1$ denotes the number of social relationships between n_i and n_j, $f_{(i,j)}$ denotes the 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 neighbors in the set of $\mathcal{S}_i$.

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 mean closer relationship. Since each node establishes its own friend-relationship network, broadcasting can be used to find the shortest paths. Basically, n_i broadcasts a message to its friends, which further broadcast the message to their friends. This process repeats until the message arrives at n_j. Then, a set of shortest paths between n_i and n_j, $\mathcal{P}_{i,j}{=}\ \{p_1, p_2, \cdots, p_k\}$, are identified. Thus, the closeness of non-adjacent nodes n_i and n_j is calculated by:

$$\Omega_{d_{(i,j)}} = \sum_{k=1}^{|\mathcal{P}_{(i,j)}|} \sum_{i=0}^{|p_k|} \Omega_{d_{(i,i+1)}}, \tag{3}$$

where $|p_k|$ denote the path length of the path p_k. That is, the social closeness between two nodes is the sum of the Ω_d between all pairs of adjacent nodes in the shortest paths.

However, broadcasting generates a large amount of overhead. Binzel *et al.* [34] 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 [35] that the users possessing a social network primarily transact with 2 to 3 hop partners. Therefore, the friend-of-friend (FOF) relationship [36] is sufficiently accurate to capture the indirect social closeness between two nodes. If two nodes have more common friends, they are more likely to have close social relationship.

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, we calculate the social closeness between n_i and n_j by:

$$\Omega_{d_{(i,j)}} = \sum_{k \in |S_i \cap S_j|} \frac{\Omega_{d_{(i,k)}} + \Omega_{d_{(k,j)}}}{2} \tag{4}$$

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 $\Omega_{(k,j)}$.

In summary:

$$\Omega_{d_{(i,j)}} = \begin{cases} \frac{m_{(i,j)} \cdot f_{(i,j)}}{\sum_{k=0}^{|S_i|} f_{(i,k)}} & n_i \text{ and } n_j \text{ are adjacent,} \\ \sum_{k \in |S_i \cap S_j|} \frac{\Omega_{d_{(i,k)}} + \Omega_{d_{(k,j)}}}{2} & n_i \text{ and } n_j \text{ are not adjacent.} \end{cases} \tag{5}$$

SocialTrust uses $\theta \bar{F}$ ($\theta > 1$) for the threshold to determine whether the rating frequency is high, where $\bar{F}$ is the average rating frequency from one node to another node in the system. For example, in Overstock, $\bar{F}$ = 2.2/month.

According to *B3* and *B4* described in Section III, when n_i rates n_j with high ratings and high frequency, if $\Omega_{d_{(i,j)}}$ 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 $\Omega_{d_{(i,j)}}$.

As shown in Figure 6, the Gaussian function is a characteristic symmetric "bell curve" shape that can mitigate or filter the effect of a factor with values greatly deviated from the normal value. It is a function of the form:

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}}, \tag{6}$$

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 $r_{(i,j)}$.

$$r_{(i,j)} = r_{(i,j)} \cdot \alpha \cdot e^{-\frac{(\Omega_{d_{(i,j)}}-\bar{\Omega}_{d_i})^2}{2|\max\Omega_{d_i}-\min\Omega_{d_i}|^2}}, \tag{7}$$

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

We set $\alpha = a$ to adjust the weight of ratings, $b = \bar{\Omega}_{d_i}$, which is the most reasonable social closeness of n_i to other nodes it has rated, and $c = |\max\Omega_{d_i} - \min\Omega_{d_i}|$, which is the greatest variance of social closeness of n_i to other nodes it has rated. The exponent in Equation (7) is the deviation of the social closeness of n_i and n_j from the normal social closeness of n_i to other nodes it has rated. We also can replace $\bar{\Omega}_{d_i}$ with the average Ω_d of a pair of transaction peers in the system based on the empirical result. For example, in Overstock, the average, maximum and minimum number of hops of a pair transaction peers are 1.54, 4 and 1.

As Figure 6 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, while 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.

Interest similarity: In SocialTrust, each node has an interest vector $\mathcal{V}$=<$v_1, v_2, v_3, ..., v_k$> indicating its interests. Each dimension v in the vector corresponds to one interest. In P2P applications, the interest vector of a peer can be derived from the resources it frequently requests or from the interests in the user's profile in social network. For example, in the P2P file sharing system, a node's interest vector can be represented by the keywords (e.g., music, sports and movie) extracted from its frequently requested files using the information retrieval algorithm [37]. Users input their interested products into their profiles in Overstock. As mentioned, the social interest similarity of n_i to n_j is calculated by:

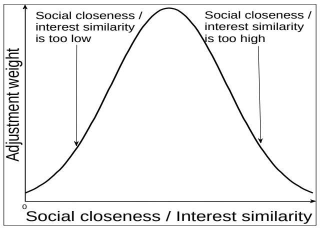


Figure 6: One-dimensional reputation adjustment.

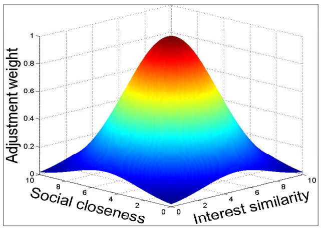


Figure 7: Two-dimensional reputation adjustment.

$$\Omega_{c(i,j)} = \frac{|\mathcal{V}_i \cap \mathcal{V}_j|}{\min(|\mathcal{V}_i|, |\mathcal{V}_j|)}. \tag{8}$$

Nodes with larger Ω_c 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 [21]. This was confirmed in previous study [38] on peoples' relations based on their interested files. P2P resource sharing and transactions usually occur between nodes sharing similar interests. For example, computer science students often search computer science related information and may only search politics related information occasionally. Hence, if two nodes n_i and n_j sharing few interests (i.e., small $\Omega_{c(i,j)}$) rate each other frequently, they are likely to collude with each other, as indicated in *B3* in Section III. On the other hand, 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 colluder.

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

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

where $\max\Omega_{c_i}$, $\min\Omega_{c_i}$ and $\bar{\Omega}_{c_i}$ 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 (9) when n_i frequently rates n_j with high ratings and $(\Omega_{c_{(i,j)}} - \bar{\Omega}_{c_i}) < 0$ which implies that n_i and n_j share few interests, or when n_i frequently rates n_j with low ratings and $(\Omega_{c_{(i,j)}} - \bar{\Omega}_{c_i}) > 0$ which implies that n_i and n_j share many interests.

Similar to social closeness, we also can replace $\bar{\Omega}_{c_i}$ with the average Ω_c 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.

A. Combination of social closeness and similarity

Combining Formulas (7) and (9), we get:

$$r_{(i,j)}(\Omega_d, \Omega_c) = r_{(i,j)} \cdot \alpha \cdot e^{-\left(\frac{(\Omega_{d_{(i,j)}}-\bar{\Omega}_{d_i})^2}{2|\max\Omega_{d_i}-\min\Omega_{d_i}|^2}+\frac{(\Omega_{c_{(i,j)}}-\bar{\Omega}_{c_i})^2}{2|\max\Omega_{c_i}-\min\Omega_{c_i}|^2}\right)}, \tag{10}$$

which simultaneously considers social closeness and interest similarity. For example, for two low-reputed nodes rating each other with high frequency, if they have very close social relationship (i.e., high $\Omega_{d_{(i,j)}}$) but share few common interests (i.e. low $\Omega_{c_{(i,j)}}$), they are more likely to collude with each other. This is because two nodes have low probability to frequently request resource from each other if they share few common interests, and a node is unlikely to request the resource from a low-reputed node. Let us use H_d and L_d to denote very high and low social closeness, and use H_c and L_c to denote very high and low interest similarity, as Figure 7 shows, the rating values between the nodes that have (H_d, H_c), (H_d, L_c), (L_d, H_c) and (L_d, L_c) are greatly reduced. Therefore, based on the Formula (10), the influences of the collusion listed in *B1-B4* are reduced.

In reputation systems, 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 calculating node global reputation periodically. Suppose M_j is the resource manager of n_j. M_j keeps the interest vector 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, if M_j does not have interest vector and friendlist of rater n_i, it contacts n_i's reputation manager M_i for the information. Based on the calculated $\Omega_{d_{(i,j)}}$ and $\Omega_{c_{(i,j)}}$ 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 $\Omega_{d_{(i,j)}}$ and $\Omega_{c_{(i,j)}}$ to represent the degree of social closeness and interest similarity between a pair of nodes, denoted by T_{d_h}, T_{d_l}, T_{c_h} and T_{c_l}, respectively. If $t^+_{(j,i)}{>}T^+_t$, which means n_j also frequently rates n_i with positive ratings, then if (1) their social closeness is low ($\Omega_{d_{(i,j)}}{<}T_{d_l}$) (*B1*), (2) their social closeness is high ($\Omega_{d_{(i,j)}}{>}T_{d_h}$) and n_j is a low-reputed node ($R_j{<}T_R$) (*B2*), or (3) their interest similarity is low ($\Omega_{c_{(i,j)}}{<}T_{c_l}$) (*B3*), M_i adjusts $r_{(i,j)}$ according to Equation (10). If $t^-_{(i,j)}{>}T^-_t$, which means n_i frequently rates n_j with negative ratings, then if their interest similarity is high ($\Omega_{c_{(i,j)}}{>}T_{c_l}$) (*B4*), M_i adjusts $r_{(i,j)}$.

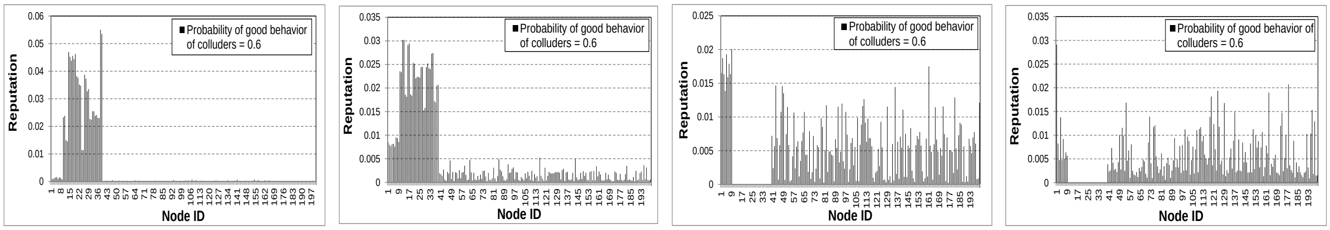
V. Performance Evaluation

Network model. We built an unstructured P2P network with 200 nodes. Our real trace shows that the total number of product categories in Overstock is around 20, and the range of the number of interests of each node is [1,10]. Thus, the number of total interests in the P2P network was set to 20, and the number of interests of each node was randomly chosen from [1,10]. Nodes with the same interest are connected with each other, and a node requests resources (resource and service are interchangeable terms in this section) from its neighbors having the interest of the requested resource. As observed in Section III, the frequency a node requests resources in its interests in the experiments conforms to a power law distribution. Each node can handle 50 requests simultaneously per query cycle. When selecting a server 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 is subdivided into 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] in each query cycle. 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 in the paper.

Node model. We consider three types of nodes: pretrusted nodes, malicious colluders and normal nodes. The pretrusted nodes always provide authentic resources to the requesters. Normal nodes provide inauthentic resources with a probability of 0.2. 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 [8, 9], 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. In order to show the experimental results clearly, we used IDs 1-9 for the pretrusted nodes and used IDs 10-39 for the colluders. We assigned the social distance between colluders to 1. Considering that 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].

Collusion model. We consider positive ratings among colluders in the experiments. Similar results can be obtained for the collusion of negative ratings. Among the colluders, the nodes receiving ratings from other nodes are called *boosted nodes* and the nodes rating others are called *boosting nodes*. We consider two major collusion models in P2Ps [8]: pair-wise collusion model (PCM) and multiple node collusion model (MCM). 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 number of boosting nodes rate a single boosted node with high



(a) EigenTrust.

(b) eBay.

(c) EigenTrust employing SocialTrust.

(d) eBay employing SocialTrust.

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

frequency in order to boost the reputation of that node, but the boosted nodes does not rate boosting nodes back.

Reputation model. In the simulated P2P network, the initial reputation of each node is 0. A client gives a service rating 1 when it receives an authentic service and rating -1 when it receives an inauthentic service. Each node's global reputation is updated once after each simulation cycle. The parameter α in the Gaussian function was set to one. We measured the performance of three reputation systems: EigenTrust, eBay and SocialTrust.

In EigenTrust, each peer n_i maintains the number of satisfactory and unsatisfactory transactions it has had with peer n_j, denoted by $sat(i, j)$ and $unsat(i, j)$, respectively. n_i calculates the local trust value of n_j: $s_{ij} = sat(i, j) - unsat(i, j)$, and normalizes the value by $c_{ij} = \frac{max(s_{ij},0)}{\sum_k max(s_{ik},0)}$. Then, we obtain a matrix C containing the trust value of the peer pairs c_{ij} in the system. $\overrightarrow{c_i}$ is a vector that stores all the local trust values that node n_i gives to all other nodes in the system. The trust vector $\overrightarrow{t_i}$ of node i is the left principal eigenvector of C; $\overrightarrow{t_i} = C^T\overrightarrow{c_i}$. In this step, the nodes with higher reputation have higher reputation rating weights. In $\overrightarrow{t_i}$, the element t_i is peer n_i's global reputation. In order to prevent collusion, $\overrightarrow{t_i} = (1-\eta)C^T + \eta\overrightarrow{p}$, where $\overrightarrow{p}$ includes the ratings from pretrusted nodes, and $\alpha \in [0, 1]$. We set η=0.5 in our experiments.

In eBay, in order to thwart collusion, multiple positive or negative ratings from node n_i to node n_j within the same week only increase or decrease n_j's reputation by one point, respectively. If a seller receives more negatives than positives from the same buyer in the same week, the seller's reputation is lowered by 1 point. If a seller receives more positives than negatives from the same buyer in the same week, the seller's reputation is raised by 1 point. In our simulation, 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.

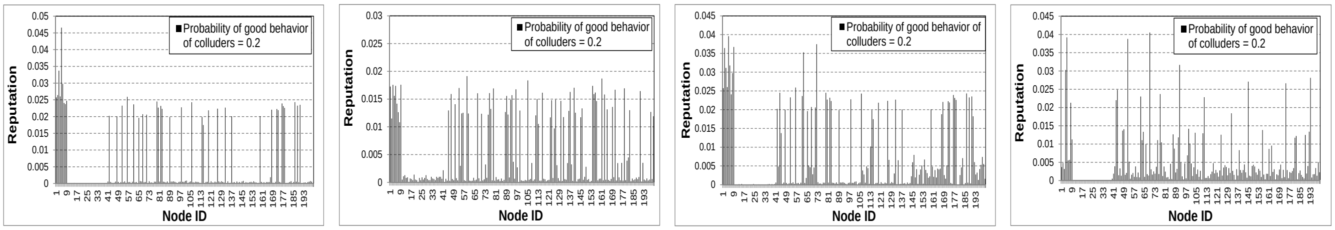
A. Effectiveness in combating pair-wise collusion (PCM)

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 the system in EigenTrust. We can see that colluders with IDs in 10-39 have much higher reputations than all other nodes. Also, the reputations of pretrusted nodes with IDs in 1-9 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. Though 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 trustworthiness of nodes 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 gain reputations. Meanwhile, the mutual positive ratings between colluders further boost their own reputations. Therefore, eBay also has low effectiveness in combating collusion and its generated reputations cannot truly reflect the trustworthiness of nodes.

Comparing Figure 8(a) and Figure 8(b), we find that 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. 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. Thus, eBay constrains reputation increase caused by collusion, leading to much lower reputations of colluders. As a result, the pretrusted and normal nodes have more opportunities to receive requests, gaining higher reputations than those in EigenTrust.

Figures 8(c) and (d) show the reputation distributions of the nodes in EigenTrust and eBay employing SocialTrust, respectively. We can see that the colluders with IDs in 10-39



(a) EigenTrust.

(b) eBay.

(c) EigenTrust employing SocialTrust.

(d) eBay employing SocialTrust.

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

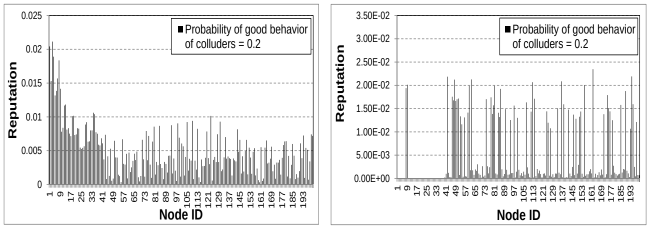
in both figures have much lower reputation values than those in Figures 8(a) and (b). The results show that SocialTrust can help EigenTrust and eBay to effectively thwart collusion. SocialTrust identifies suspected colluders based on social closeness and distance, and adjusts their reputation. Thus, the colluders in SocialTrust 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 colluders provide authentic services with 0.2 probability (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. Though colluders rate each other frequently, the weight of their ratings are very low due to their low-QoS and low reputations. Thus, they finally receive low reputations, and hence few service requests. As a result, the normal and pretrusted nodes have more opportunities to raise their reputations. Since the pretrusted nodes with IDs in 1-9 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 one from a number of options with the same reputation value 0. Since the chosen node earns reputation and subsequently has higher probability to be chosen. Therefore, EigenTrust can counter collusion when the colluders offer low-QoS at most of the time.

Figure 9(b) shows the reputation distribution of nodes in eBay. The reputations of colluders are much lower than those of the pretrusted nodes and normal nodes. The colluders receive low ratings from normal nodes due to their high probability of misbehaving. Though the colluders rate each other with high frequency in order to boost their reputations, as eBay disregards the ratings from the same rater in the same simulation cycle, their final reputation values are still very low. Because colluders still receive high ratings with 0.2 probability and these ratings are not adjusted by weight, they earn slightly higher reputations than in EigenTrust.

Figures 9(c) and (d) show the reputation distribution of nodes in EigenTrust and eBay employing SocialTrust, respectively. Both figures show that the reputation values of colluders are nearly 0. By considering social closeness and interest relationship between the nodes, SocialTrust reduces the impacts of the ratings from the potential colluders, and thus reduces the reputation values of the colluders.

B. Effectiveness in combating pair-wise collusion (PCM) with compromised pretrusted nodes



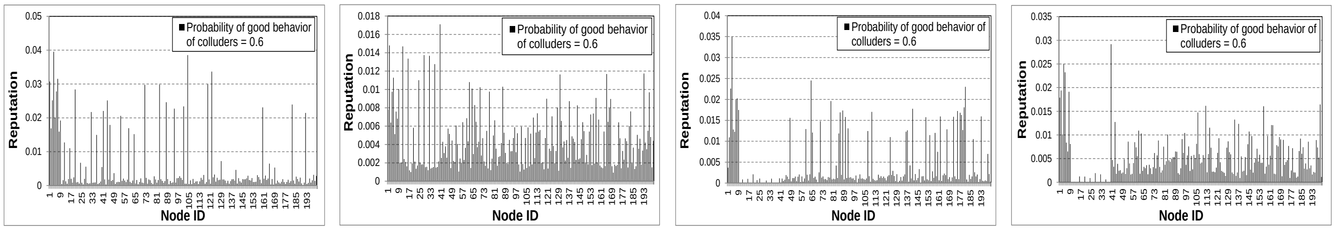
(a) EigenTrust.

(b) EigenTrust employing SocialTrust.

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

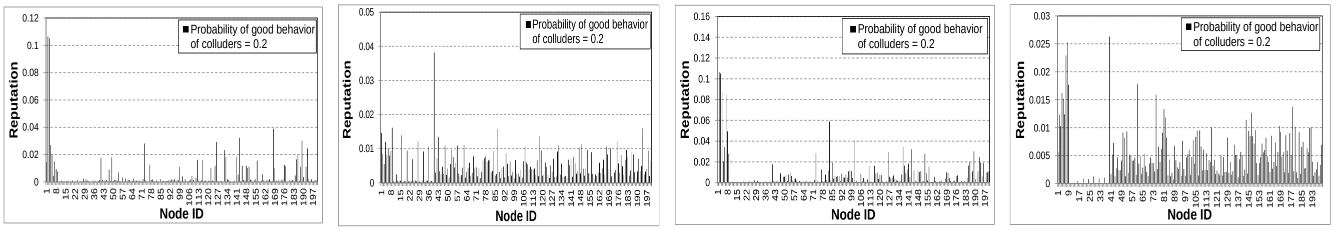
The previous experimental results show that EigenTrust is effective in combating colluders when B=0.2, but not effective when B=0.6. Next, we consider a scenario where B=0.2 and compromised pretrusted nodes are involved in the collusion. We randomly select 7 nodes from the pretrusted nodes and let them randomly select a colluder to collude with. 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 the collusion involvement of pretrusted nodes greatly boosts the reputations of themselves and colluders, and reduces the reputations of normal nodes accordingly. This is due to three reasons. First, the ratings of pretrusted nodes have higher weight and they rate highly on the colluders, the reputations of the colluders in collusion with the pretrusted nodes are increased. 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.



(a) EigenTrust. *(b)* eBay. *(c)* EigenTrust employing SocialTrust. *(d)* eBay employing SocialTrust.

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



(a) EigenTrust. *(b)* eBay. *(c)* EigenTrust employing SocialTrust. *(d)* eBay employing SocialTrust.

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

Figure 10(b) shows the reputation distribution of the nodes in EigenTrust employing 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, even though 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.

C. Effectiveness in combating multiple node collusion (MCM)

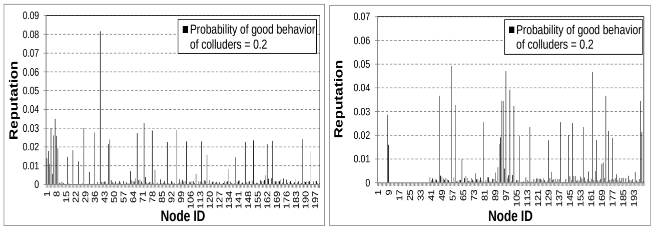
In the multiple node collusion model, among the 30 colluders, 7 nodes are randomly selected as the boosted nodes, and all other colluders randomly select one of the boosted nodes to collude with. We first set the probability that colluders provide authentic services to 0.6 (B=0.6).

Figure 11(a) shows the reputation distribution of nodes in EigenTrust. 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 node receive fewer and fewer requests, thus having reduced opportunities to raise their reputation values.

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). Comparing Figure 11(a) and Figure 11(b), we find the reputation values of the boosted nodes in EightTrust are much higher than those in eBay. The reason is the same as in Figures 8(a) and (b).

Figure 11(c) plots the reputation distribution of nodes in EigenTrust employing 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, as the values of these ratings are reduced according to the social and interest relationship between the raters and ratees, the overall reputation values of those boosted nodes do not increase significantly. 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. As the boosted node can provide authentic services with probability of 0.6, the boosted nodes still have low reputation values.

Figure 11(d) shows the reputation distribution of the nodes in eBay employing SocialTrust. It shows that SocialTrust can effectively fight against collusion. Although the boosting



(a) EigenTrust in MCM.

(b) EigenTrust employing SocialTrust in MCM.

Figure 13: Reputation distribution in MCM with compromised pretrusted node with B=0.2 (pretrusted nodes: 1-9, colluders: 10-39).

nodes can increase the reputation values of the boosted nodes as shown in Figure 11(b), since the SocialTrust reduces the impact of the rating between the colluders based on their social closeness and interest similarity, 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 and measured the performance of different systems. Figure 12(a) shows the reputation distributions of the nodes in EigenTrust. It shows that the reputations of the colluders including 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. Thus, 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. Therefore, 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 some of 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 employing SocialTrust, respectively. The figures show that SocialTrust further reduces the reputation values of the boosted nodes. The results demonstrate the effectiveness of SocialTrust in reducing the impact of the ratings from colluders on node reputations by considering their social and interest relationships.

D. Effectiveness in combating multiple node collusion (MCM) with compromised pretrusted nodes

Figure 13(a) demonstrates the reputation distribution of the nodes in EigenTrust in MCM, 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 13(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 hence low weight for their ratings. Thus, 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 the pretrusted node are compromised, as shown in Figure 13(a), their ratings greatly increase the reputations of the boosted nodes, which attract many requests from the pretrusted nodes.

Figure 13(b) shows the reputation distribution of the nodes in EigenTrust employing SocialTrust in MCM. 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, which confirms the capability of SocialTrust in countering collusion. The pretrusted nodes with IDs in 8-9 have very high reputations because they are not involved in collusion.

E. Percentage of requests sent to colluders

Table I 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 enhances 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 PCM than MCM. This is because colluders in PCM mutually rate each other with high frequency, while

Table I: 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%

boosting nodes in MCM do not receive ratings from boosted nodes. As a result, all colluders in PCM have high reputations and attract more service queries. While in MCM, the reputation values of boosting nodes are very low especially with B=0.2. Thus, the weight of their ratings is small, which cannot significantly increase the reputation values of the boosted nodes. With relatively lower reputations, the colluders cannot attract as many requests as in PCM.

Third, we see that 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 can reduce 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. Thus, these nodes receive low reputations and fewer service requests, which discourages the collusion behaviors.

VI. Conclusion

Despite the effectiveness of reputation systems in finding deceptive peers according to the reputation values, they are vulnerable to collusion. Though 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 social network to combat collusion. Experimental results show that SocicalTrust greatly enhances the capability of eBay reputation system and EigenTrust in countering collusion. SocicalTrust can even detect colluders when compromised pretrusted high-reputed nodes are involved in collusion. In our future work, we will further investigate how to determine appropriate thresholds used in this paper.

Acknowledgements

This research was supported in part by U.S. NSF grants OCI-1064230, CNS-1049947, CNS-1025652, CNS-1025649, and CNS-0917056, Microsoft Research Faculty Fellowship 8300751, and Sandia National Laboratories grant 10002282.

References

[1] Bittorrent. http://en.wikipedia.org/wiki/Bittorrent.

[2] Gnutella home page. http://www.gnutella.com.

[3] PPLive. http://www.pplive.com.

[4] M. Cai, M. Frank, J. Chen, and P. Szekely. MAAN: A Multi-Attribute Addressable Network for Grid Information Services. *Journal of Grid Computing*, 2004.

[5] eBay. http://www.ebay.com.

[6] Amazon. http://www.amazon.com/.

[7] Overstock. http://www.overstock.com/.

[8] 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.

[9] S. Zhao and V. Lo. Result verification and trust-based scheduling in open peer-to-peer cycle sharing systems. In *Proc. of P2P*, 2005.

[10] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In *Proc. of WWW*, 2003.

[11] The online marketplace: eBay. http://www.ebay.com.

[12] 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.

[13] M. Srivatsa, L. Xiong, and L. Liu. Trustguard: Countering vulnerabilities in reputation management for decentralized overlay networks. In *Proc. of WWW*, 2005.

[14] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust incentive techniques for peer-to-peer networks. In *Proc. of EC*, 2004.

[15] Z. Liang and W. Shi. Pet: A personalized trust model with reputation and risk evaluation for p2p resource sharing. In *Proc. of HICSS*, 2005.

[16] R. Sherwood S. Lee and B. Bhattacharjee. Cooperative peer groups in nice. In *Proc. of INFOCOM*, 2003.

[17] A. A. Selcuk, E. Uzun, and M. R. Pariente. A reputation based trust management system for P2P networks. *IJNS*, 2008.

[18] 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.

[19] 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.

[20] N. Curtis, R. Safavi-Naini, and W. Susilo. X^2Rep: Enhanced trust semantics for the XRep protocol. In *Proc. of ACNS*, 2004.

[21] M. Mcpherson. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

[22] L. Xiong and L. Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *TKDE*, 16(7), 2004.

[23] A. Singh and L. Liu. Trustme: Anonymous management for trust relationships in decentralized p2p systems. In *Proc. of P2P*, 2003.

[24] M. Schlosser S. Kamvar and H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In *Proc. of WWW*, 2003.

[25] R. Zhou and K. Hwang. Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE TPDS*, 2007.

[26] S. Song, K. Hwang, R. Zhou, and Y.-K. Kwok. Trusted P2P transactions with fuzzy reputation aggregation. *IEEE Internet Computing*, 2005.

[27] R. Zhou and K. Hwang. Gossip-based reputation management for unstructured peer-to-peer networks. *IEEE TKDE*, 2007.

[28] C. P. Costa and J. M. Almeida. Reputation systems for fighting pollution in peer-to-peer file sharing systems. In *Proc. of P2P*, 2007.

[29] K. Walsh and E. Sirer. Experience with an object reputation system for peer-to-peer file-sharing. In *Proc. of NSDI*, 2006.

[30] F. Cornelli, E. Damiani, S. Vimercati, S. Paraboschi, and P. Samarati. Choosing reputable servents in a P2P network. In *Proc. of WWW*, 2002.

[31] T. Moreton and A. Twigg. Trading in trust, tokens, and stamps. In *Proc. of P2PEcon*, 2003.

[32] A. Fast, D. Jensen, and B. N. Levine. Creating social networks to improve peer to peer networking. In *Proc. of KDD*, 2005.

[33] H. Yu, M. Kaminsky, P. B. Gibbons, and A. D. Flaxman. SybilGuard: defending against sybil attacks via social networks. *TON*, 2008.

[34] C. Binzel and D. Fehr. How social distance affects trust and cooperation: Experimental evidence in a slum. In *Proc. of ERF*, 2009.

[35] 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.

[36] S. Garriss, M. Kaminsky, M. J. Freedman, B. Karp, D. Mazieres, and H. Yu. Re: Reliable Email. In *Proc. of NSDL*, 2006.

[37] M.W. Berry, Z. Drmač, and E . R. Jessup. Matrices, vector spaces, and information retrieval. *SIAM review*, 1999.

[38] A. Iamnitchi, M. Ripeanu, and I. Foster. Small-world file-sharing communities. In *Proc. of INFOCOM*, 2004.