Developing Trust in Large-Scale Peer-to-Peer Systems

Bin Yu
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
byu@cs.cmu.edu

Munindar P. Singh
Department of Computer Science
North Carolina State University
Raleigh, NC 27695, USA
singh@ncsu.edu

Katia Sycara
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
katia@cs.cmu.edu

Abstract

In peer-to-peer (P2P) systems, peers often must interact with unknown or unfamiliar peers without the benefit of trusted third parties or authorities to mediate the interactions. A peer will need reputation mechanisms to incorporate the knowledge of others to decide whether to trust another party in P2P systems. This paper discusses the design of reputation mechanisms and proposes a novel distributed reputation mechanism to detect malicious or unreliable peers in P2P systems. It illustrates the process for rating gathering and aggregation and presents some experimental results to evaluate the proposed approach. Moreover, it considers how to effectively aggregate noisy (dishonest or inaccurate) ratings from independent or collusive peers using weighted majority techniques. Furthermore, it analyzes some possible attacks on reputation mechanisms and shows how to defend against such attacks.

Keywords P2P security, reputation mechanisms, trust

1. Introduction

A major challenge for large-scale P2P systems is how to establish trust between different peers without the benefit of trusted third parties or authorities. Usually the peers don't have any pre-existing relationship and may reside in different security domains. Sometimes even when there are some authorities available, e.g., an authentication server or certification authority, it is inadvisable to assume that these authorities can monitor transactions and then declare the trustworthiness of different peers. The research of trust in security focuses on creating, acquiring, and distributing certificates [1]. A conventional certificate chain, even if perfect and not compromised, would at best attest to the identity of the given party, but would not be able to guarantee that the given party is in fact trustworthy for a particular purpose at hand, e.g., making a small payment or signing a million-dollar purchase order [8].

Consequently, peers must rely on reputation mechanisms for deciding whether to trust another party based on its past history. Reputation mechanisms are about generation, discovery, and aggregation of rating information in electronic commerce and P2P systems. Online reputation systems like eBay and Amazon have been designed to foster trust among strangers in electronic commerce [19]. However, most existing online reputation systems are centralized and may not be compatible with the design philosophy of P2P systems. Some researchers have begun to examine reputation-based approaches in P2P systems, where peers keep track of and share the rating information about each other [5, 6, 11, 15]. The distributed polling algorithms in these approaches are based on Gnutella protocol by which requesters access the reliability of perspective providers. These frameworks are mainly designed for reducing the spread of malicious programs in P2P file sharing systems like Gnutella. Most of them use binary ratings and do not consider the efficiency of polling algorithms and noisy ratings from dishonest or unreliable peers.

This paper proposes a distributed reputation mechanism for P2P systems in general, e.g., multiagent systems (each peer is a software agent) [10], and the web services (each peer is a web service provider) [16], where binary ratings cannot accurately model a peer's experience of the *quality of service* (QoS) with other peers. This paper focuses on the design of reputation mechanisms on unstructured P2P systems, and does not consider structured P2P systems with Distributed Hash Tables (DHTs), e.g., CAN [17] and Chord [21]. One reason is that DHTs are mainly designed for distributed storage systems, while the high turnover rate caused by frequent join and leave of peers in dynamic P2P systems causes significant overhead for DHTs [4].

This paper goes beyond existing approaches in the following three ways.

Ratings Generation: The ratings in existing approaches are binary. In the binary ratings, a peer rates the services from another peer as one of two values, commonly interpreted as either one (e.g., positive or satisfactory) or zero

(e.g., negative, unsatisfactory). Binary ratings work pretty well for file sharing systems where a file is either the definitive correct version or is wrong, but cannot accurately model richer services in other settings such as web services and electronic commerce, where a boolean may not adequately represent a peer's experience of the *quality of service* (QoS) with other peers, e.g., the quality of products the peer sends and the expected delivery time. Our approach considers quality of service (QoS) as probabilistic ratings in the interval $[0, 1]$ and focuses on how to aggregate these ratings.

Ratings Discovery: The polling algorithms for ratings discovery are based on Gnutella protocols, in which the requesting peer broadcasts the message to all other peers within the horizon of a given TTL (Time to Live). Polling processes waste much bandwidth and processing power since each peer queries all of its neighbors. Our approach applies a process of referrals through which peers help one another find witnesses [12, 25]. The process of referrals requires that any referrals, e.g., the names and addresses of other peers, are sent back to the requesting peer. Our approach yields a better performance compared with polling algorithms, where peers only send queries to a subset of their neighbors.

Ratings Aggregation: Although some of the existing approaches consider the credibilities of voters (or witnesses) in the enhanced polling protocol, they don't consider how to effectively aggregate the noisy ratings in presence of dishonest or unreliable voters [5]. For example, how to identify deceptive or unreliable peers and how to adjust the ratings from these peers? We discuss different models of deception in the process of rating aggregation, e.g., complementary, exaggerated positive, and exaggerated negative, and study how to distinguish reliable peers from deceptive or unreliable peers. The focus of this paper is on minimizing the effect of ratings from these independent or collusive peers using weighted majority techniques.

We assume the use of Public Key Infrastructure (PKI) for naming and authentication in P2P systems. The basic idea is that each peer has a matched public key and private key. The private key is used to encrypt the message so that only peers that know the corresponding public key can decrypt the message. The goal of our research is to improve the security level of large-scale P2P systems based upon PKI and reputation mechanisms, where PKI provides the channel for secure communication and reputation mechanisms help peers detect malicious or unreliable peers and lead to more robust and secure P2P systems.

The rest of this paper is organized as follows. Section 2 summarizes the relevant literature. Section 3 describes the design of reputation mechanisms, including generation, discovery, and aggregation of rating information. Section 4 presents some experimental results. Section 5 analyzes some attacks in reputation mechanisms, and Section 6 presents some directions for future research.

2. Literature

Cornelli *et al.* propose a reputation-based approach for P2P file sharing systems (called P2PRep) [5]. In P2PRep, a peer pools other peers by broadcasting a request about the opinion of the select peer. Damiani *et al.* present a similar approach, called XRep, which considers the reputations of both peers and resources [6]. P2PRep and XRep do not give any metrics to quantify the credibilities of voters. Also, they only can find malicious peers within a given horizon. Our approach involves an adaptive process of neighbor selection, which may help to detect malicious peers who are originally beyond the horizon.

Kamvar *et al.* propose a reputation-based approach, called EigenRep, for P2P file sharing systems [11]. In EigenRep, each peer is assigned a unique global reputation value, which is computed using an algorithm similar to PageRank [2]. However, it is not clear if their approach is feasible for large-scale P2P systems, in which some local reputation values are unreachable for the requesting peers. Richardson *et al.*'s approach to trust management for semantic web is similar to EigenRep, but ratings are personalized for each user based on her personal experience [20]. Both approaches simply assume that peers are honest and therefore cannot defend some attacks like deceptions and rumors.

More recently, Marti and Garcia-Molina [15] discuss the effect of reputation information sharing on the efficiency and load distribution of a peer-to-peer system, in which peers only have limited (peers share their opinions) or no information sharing (peers only use their local ratings). In their approach, each node records ratings of any other nodes in a reputation vector of length n, where n is the total number of nodes in the network. Their approach does not distinguish the ratings for service (reliability) and ratings for voting (credibility) and does not consider how to adjust the *weight* for votings with the number of local ratings.

This paper is related to our previous approach to distributed reputation management in multiagent systems [23, 24]. In our previous approach we adapted the mathematical theory of evidence to represent the ratings that agents give to their correspondents. We also discussed how to distinguish reliable witnesses from deceptive witnesses through a variant of weight majority algorithm. However, our previous approach does not consider the difference between simple averaging and exponential averaging for local ratings and does not address attacks from a colluding group. This paper focuses on the design of robust and efficient reputation mechanisms in P2P systems and studies possible attacks of reputation mechanisms in P2P systems.

The research of trust in security focuses on creating, acquiring, and distributing certificates. i.e., whether to authorize a request between any parties that know little each other, or how to formulate security policies and determine whether particular sets of certificates satisfy the relevant policies [1]. The notion of trust in security is based on access control, and assumes that trust is equivalent to delegation [13, 18]. The purpose of trust here is to provide protection from those who offer services, rather than from those who want to access them (in P2P systems) [6].

Trust negotiation is an approach to establishing trust through the use of access control policies [26]. The policies specify what kinds of credentials a stranger must disclose to have access to a local resource. Yu *et al.* study the different strategies of credentials exchange, i.e., the length of the process and the amount of information disclosed. Their approach belongs to the traditional trust management in security (property-based authentication and authorization systems), and doesn't address any reputation issues.

3. Reputation Mechanisms

In P2P systems peers form ratings of others that they interact with. To evaluate the trustworthiness of a given party, especially prior to any frequent direct interactions, the peers must rely on incorporating the knowledge of other peers—termed *witnesses* —who have interacted with the same party using reputation mechanisms. In our framework, each peer has a set of *acquaintances*, a subset of which are identified as its *neighbors*. The neighbors are the peers that the given peer would contact and the peers that it would refer others to. A peer maintains a model of each acquaintance. This model includes the acquaintance's *reliability* to provide high-quality services and *credibility* to provide trustworthy ratings to other peers. More importantly, peers can adaptively choose their neighbors based on the average of local ratings, which they do every so often from among their current acquaintances, e.g., every 5 queries for a peer.

3.1. Local and Aggregate Ratings

Our approach considers the quality of service (QoS) from a peer as a probabilistic rating in the interval $[0, 1]$ and focuses on how to effectively aggregate these ratings. When peer P_i is evaluating the trustworthiness of peer P_j from a group of potential partners, there are two components to the evidence: the services offered directly by peer P_j and the testimonies from other peers in case P_i has had no frequent transactions with P_j before.

Local Rating A peer's local rating about another peer is based on its direct interactions with the second peer. The local rating is generated every time when an interaction takes place. Suppose peer P_i has rated the quality of service of the latest h interactions with P_j as a series of probabilistic ratings, $S_{ij} = \{s_{ij}^1, s_{ij}^2, \ldots, s_{ij}^h\}$, where $0 \leq s_{ij}^k \leq 1$, and h is bounded by the allowed history H. The local rating or the *reliability* of peer P_i for P_j can be computed as the following two ways.

(1) simple averaging

$$R(P_i, P_j) = \begin{cases} \sum_{k=1}^{h} s_{ij}/h & h \neq 0 \\ 0 & h = 0 \end{cases} \quad (1)$$

(2) exponential averaging

$$R(P_i, P_j) = \begin{cases} \gamma[s_{ij}^h + \ldots + (1-\gamma)^h s_{ij}^1] & h \neq 0 \\ 0 & h = 0 \end{cases} \quad (2)$$

where γ $(0 < \gamma < 1)$ is the averaging constant and determines the weights given to the most recent past observations. The bigger the γ is, the faster the past observation is forgotten. The simple averaging and exponential averaging have similar results if the peers behave in a consistent manner. However, the estimate of the current rating in the simple averaging will tend to lag behind the true value of the current rating for a malicious peer P_j if P_j is exploring the reputation mechanisms. For example, the simple averaging is not sensitive to the attacks of (malicious) peers, where peers may accumulate a high reputation and then attack the P2P systems.

Aggregate Rating A peer's aggregate rating about another peer combines the local ratings (if any) with testimonies received from any witnesses. Aggregate rating can be used for deciding whether the other peer is trustworthy and cannot be propagated to other peers. Suppose $\{W_1, \ldots, W_L\}$ are a group of witnesses towards peer P_j and the testimony $R(W_k, P_j)$ is witness W_k's local rating for peer P_j, w_k is the weight for the *credibility* of witness W_k, then the prediction from the testimonies is

$$\mathcal{P} = \begin{cases} \sum_{k=1}^{L} w_k * R(W_k, P_j)/L & L \neq 0 \\ 0.5 & L = 0 \end{cases} \quad (3)$$

The aggregate rating towards peer P_j is

$$T(P_i, P_j) = \begin{cases} \eta R(P_i, P_j) + (1-\eta)\mathcal{P} & L \neq 0 \\ 0.5 & L = 0 \end{cases} \quad (4)$$

where η is peer P_i's confidence about its local rating for peer P_j and $\eta = h/H$; L is the number of witnesses found by P_i and $1 \leq k \leq L$. Equation 4 tells us that reputation mechanisms help to establish trust between peers P_i and P_j, but the ratings from their direct interactions become more and more important in deciding whether peer P_i trusts peer P_j. When Peer P_i has not interacted with peer P_j before ($h = 0$), peer P_i has to rely on the testimonies from

other peers to decide whether to trust peer P_j. If the aggregate rating from testimonies (local ratings are empty) is above a threshold ω_i, peer P_i will interact with peer P_j. Otherwise, peer P_i marks peer P_j as *unreliable* and will not interact with it in the future. When $h = H$, peer P_i decides whether to trust peer P_j solely based on its direct interactions and will not rely on the testimonies from other peers. [1] The aggregate rating $T(P_i, P_j)$ for a new peer P_j is equal to 0.5 in our design, where $R(P_i, P_j) = 0$ and $L = 0$. The idea is that suspicion of new peers is socially inefficient since malicious peers are rare in the P2P system [9]. In a P2P system where peers join and leave the system dynamically, it would be more efficient to trust new peers until they are proved untrustworthy.

3.2. Ratings Discovery

One challenging problem in reputation mechanisms is how to find the right witnesses in an efficient manner? In the polling algorithm, the requesting peer broadcasts a request to its neighbors, who propagate the request to all their neighbors , and so on. If the request is matched, e.g., a witness is found, a reply is sent back following the reverse path of the request. A node will ignore the request if the node finds it comes from the same requesting peer and it is about the trustworthiness of the same peer.

Our approach applies a process of referrals through which peers help one another find witnesses. The process of referrals requires that any referrals are sent back to the requesting peer. The generation of referrals is based on the credibility of peers. Each peer may specify a threshold σ_i. The threshold can be adjusted to tune the number of referrals that the peer will give to others. The maximal number of referrals can be generated by each peer is called *branching factor* B. Note that the number of referrals generated by each peer in our approach is usually less than the number of neighbors a peer has.

Definition 1 Formally, a referral r to peer P_j returned from peer P_i is written as $\langle P_i, P_j \rangle$. Here we say P_i is a *parent* of P_j and P_j is a *child* of P_i.

A series of referrals makes a referral chain as $\langle P_i, P_{i+1}, \ldots, P_{i+l} \rangle$, where l is the length of the referral chain, and peer $i + l$ is one of the witnesses. Then *ancestor* and *descendant* are easily defined based on parent and child, respectively. The referral chains for different witnesses induce a directed graph—termed *trust graph*—whose root is the requesting peer. The *depth* of a referral is its distance on the shortest path from the root.

1 In practice, P_i may query other peers about the ratings of P_j again if the time interval of last interaction with P_j is long. This may require a certain amount of communication bandwidth, but this will prevent attacks from some strategic peers, who may first accumulate high ratings in the community and then start to attack the system.

Definition 2 A trust graph $(P_r, P_g, \mathbf{P}, \mathbf{R})$ is a directed graph, built from the referral chains produced from P_r's query about the trustworthiness of P_g, where $\mathbf{P}$ is a finite set of peers $\{P_1, \ldots, P_N\}$, and $\mathbf{R}$ is a set of referrals $\{r_1, \ldots, r_n\}$.

Algorithm 1 Constructing a trust graph

```
Suppose peer P_r is evaluating the trustworthiness of
peer P_g and P is the set of peers being visited.
for (any peer P_i ∈ P and P_i has not been queried) do
  if (depth(P_i) < depthBound) then
    P_r queries P_i
    if (P_i is a witness of P_g) then
      P_i returns the rating about P_g to P_r
    else
      for (any referral r = ⟨P_i, P_j⟩ from P_i) do
        if (P_j ∉ P) then
          Add r into R and Add P_j into P
        else
          Ignore referral r
        end if
      end for
    end if
  end if
end for
```

In this paper we only consider the trust graphs as trees and we assume peers are always willing to disclose their ratings to the requesting peer. Algorithm 1 summarizes the process of constructing a trust graph, where depthBound is the bound of referral chains in the trust graph. In the context of Figure 1, P_0 tries to evaluate the trustworthiness of P_8 and P_4 and P_7 are two witnesses for P_8. The requesting peer P_0 is black; the peers that have been queried are gray; the peers who have not been queried are white.

The differences between polling algorithms and trust graphs can be summarized as follows,

- The trust graph is constructed by the requesting peer P_r and it is only local to peer P_r. The requesting peer can adaptively direct or end the process. In polling algorithms, the messages continue to propagate until the TTL of messages is reached. The requesting peer can specify the value of TTL, but it cannot control when the process stops.
- Instead of sending queries to all neighbors, a peer in a trust graph only sends referrals to a subset of its neighbors. A trust graph usually yields a better performance compared with the polling algorithm. In the worse case, a trust graph will cost the same number of messages (each referral is a message) as a polling algorithm if the requesting peer specifies the same depth-

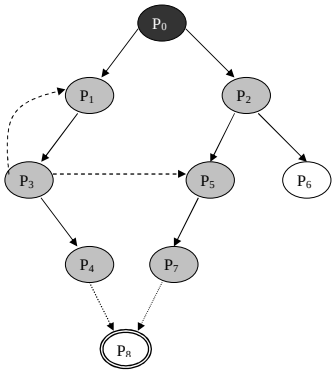

Figure 1. A trust graph generated from a query of P_0.

Bound as TTL (in the polling algorithm) and every peer returns all of its neighbors as referrals.

- Poll algorithms preserve the anonymity of both requesting peers and witnesses, who provide ratings to requesting peers through the intermediate peers. However, the anonymity may not be helpful for aggregating noisy ratings since the requesting peer does not know where the ratings come from and who recommends the lying witness. Instead in trust graphs there is a referral chain between a requesting peer and any witness and the referral chain helps the requesting peer decide whether the witness is trustworthy.

3.3. Noisy Ratings

In practice, the witnesses may not always reveal their true ratings about other peers. Sometimes the witnesses may exaggerate positive or negative ratings, or offer testimonies that are outright false. The reasons could be the witnesses only have limited knowledge of the peers or the witnesses simply try to exploit the reputation mechanisms. In this section we study the problem of noisy ratings as it may occur in rating aggregation, where a witness gives a rating about a given peer to the requesting peer.

Suppose peer P_i considers the latest h episodes of interaction with peer P_j, with the true ratings of $S_{ij} = \{s_{ij}^1, s_{ij}^2, \ldots, s_{ij}^h\}$, where $1 \leq h \leq H$. Now P_i can be deceptive in providing a rating of P_j to others. We consider three kinds of noisy ratings: complementary, exaggerated positive, and exaggerated negative. Below, α ($0 < \alpha < 1$) is the *exaggeration coefficient*, s is the true rating, and s' is the rating in the corresponding (deception) model.

$$s' = \begin{cases} 1 - s & \text{complementary} \\ \alpha + s - \alpha s & \text{exaggerated positive} \\ s - \alpha s/(1-\alpha) & \text{exaggerated negative} \end{cases} \quad (5)$$

The malicious peers could be *independent*: they give a bad rating of everyone else, or *in a colluding group*: they give good ratings of each other in the group and bad ratings of other peers. Here a bad rating could be an all-zero or a complementary rating. A good rating could be an all-one or an exaggerated positive rating. One example of a colluding group is that a single physical user generating multiple IDs such as at least one of his IDs gets higher rating [7]. In next section we show if reputation mechanisms can detect these two kinds of malicious peers or at least make the malicious attack costly.

We adapt the weighted majority algorithm (WMA) to predict the trustworthiness of a given party based on a set of testimonies from the witnesses. The original WMA algorithm deals with how to make an improving series of predictions based on a set of advisors [14]. The first idea is to assign weights to the advisors and to make a prediction based on the weighted sum of the ratings provided by them. The second idea is to tune the weights after an unsuccessful prediction so that the relative weight assigned to the successful advisors is increased and the relative weight assigned to the unsuccessful advisors is decreased.

Basically, each peer maintains a weight for the credibility of each of the other peers whose testimonies it requests. This weight estimates how credible the given witness is. Now suppose peer P_i wishes to evaluate the trustworthiness of peer P_j. Our algorithm is given from the perspective of P_i. Let $\{W_1, \ldots, W_L\}$ be a set of witnesses that P_i has discovered for peer P_j and $R(W_k, P_j)$ is the local rating returned by witness W_k. Let P_i assign a weight w_i to witness W_i. The weights of witnesses are initialized to 1 if they are not acquaintances of P_i, and will be updated after each interaction (if any). The prediction from the testimonies can be computed as Equation 3. If the aggregate rating from testimonies is above a threshold ω_i, P_i will interact with P_j. Assume the new service from P_j is rated as s by P_i, where $0 \leq s \leq 1$. The weight of witness W_k will be updated as $w_k = \theta w_k$, where the update factor θ is defined as

$$\beta^{|R(W_k,P_j)-s|} \leq \theta \leq 1 - (1-\beta)|R(W_k, P_j) - s| \quad (6)$$

where β is a constant and $0 < \beta < 1$. For simplicity, we choose the upper bound as the value of θ in this paper.

$$\theta = 1 - (1-\beta)|R(W_k, P_j) - s| \quad (7)$$

The value of θ is determined by the constant β and $|R(W_k, P_j) - s|$. The latter measures how far the prediction of witness W_k is from the rating s. If the value of $|R(W_k, P_j) - s|$ is big, it is likely that witness W_k is lying. For the same β, the bigger the value of $|R(W_k, P_j) - s|$, the smaller the value of θ. For example, if $\beta = 0.5$, $\theta = 0.95$ when $|R(W_k, P_j) - s| = 0.1$, and $\theta = 0.55$ when $|R(W_k, P_j) - s| = 0.9$. For both cases, the requesting peer needs to adjust the weight w_k of witness W_k using θ in Equation 7. Consequently, the testimonies from a witness will have a reduced effect on the aggregated ratings in the future if the witness is found lying.

4. Experimental Results

Our experiments are based on a simulation testbed of peer-to-peer information systems. This testbed models the *expertise* for each peer via vectors of dimension 5, which are randomly generated in the beginning. The value of e_i of an expertise vector $E = \{e_1, e_2, \ldots, e_5\}$ means the expertise level in the domain e_i. The queries correspond to vectors of length 5 that are 1 in one dimension and 0 in all other dimensions. For example, $[1, 0, 0, 0, 0]$ would be a query in the topic of e_1. The queries of each peer are randomly chosen as vectors that are 1 in one or two dimensions and 0 in all other dimensions and are used throughout. Moreover, we introduce a probability $\mathcal{Q}_i$ between 0 and 1 to model the quality of service (QoS) of each peer P_i. Peer P_i will generate an answer from his expertise vector with quality $\mathcal{Q}_i$ when there is a good match between the query and his expertise vector, e.g., $e_1 \geq 0.5$ for the query in topic e_1.

For each round, we randomly designate a peer to be the querying peer. When a peer receives a query, it may ignore the query, answer it based on its expertise vector, or refer to other peers. The originating peer collects all suggested referrals, and continues the process by contacting some of them. Finally, the referral process draws to an end if the length of referral chains reaches the bound D. For any peer P_j who claims it has the answer, the querying peer needs to decide if it should interact the peer P_j using reputation mechanisms. The querying peer aggregates the ratings based on weights it has assigned to the witnesses. The querying peer may interact with P_i if the aggregate rating is above a threshold $\omega_i = 0.5$. We assume the querying peer rates the service from P_i as $\mathcal{Q}'_i$ and $\mathcal{Q}'_i = \mathcal{Q}_i$. Depending on the outcome of the interactions $\mathcal{Q}'_i$, the querying peer adjusts the weights it assigns to the witnesses involved.

4.1. Setup

The topology of the system is initialized as a directed random graph. We use a random graph with 100 nodes, and approximate 4 out-edges per node (to its neighbors) as a starting point for the experiment. Note that the topology of the system will not be a random graph when peers adaptively choose their neighbors based on the local ratings. Our previous work shows that the random graph will converge to a small-world network from local interactions according to the two metrics, clustering coefficient and average length of shortest paths [22, 25].

The total of 100 peers can be divided into four groups: (1) $\mathcal{G}_1$ has 50 - 70 peers who always give normal ratings; (2) $\mathcal{G}_2$ has 10 - 30 peers who give complementary ratings; (4) $\mathcal{G}_3$ has 10 peers who exaggerate positive ratings (e.g., $\alpha = 0.1$); (5) $\mathcal{G}_4$ has 10 peers who exaggerate negative ratings (e.g., $\alpha = 0.1$). The malicious peers in a group could be *independent*: they give a bad rating of everyone else, or *in a colluding group*: they give true ratings of each other in the group and complementary ratings of other peers. We assume (1) peers in $\mathcal{G}_2$ are malicious and the QoS of each peer in $\mathcal{G}_2$ usually is 0.1; (2) peers in $\mathcal{G}_3$, and $\mathcal{G}_4$ are unreliable and the QoS of each peer in $\mathcal{G}_3$ and $\mathcal{G}_4$ is 0.5.

The peers are limited to having no more than 4 neighbors and 16 acquaintances. Queries are sent only to and referrals are given only to neighbors. After every 5 queries, each peer decides which acquaintances are promoted to become neighbors and which neighbors are demoted to be ordinary acquaintances based on the latest ratings. Other parameters are defined as follows,

Symbol	Value	Description
h	dynamic	Number of latest interactions
H	10	Bound of the allowed history
D	4	Bound of the referral chain's length
B	2	Branching factor
α	0.1	Exaggeration coefficient
β	0.5	Constant
γ	0.5	Averaging constant
θ	-	Update factor in Equation 7
η	h/H	Confidence about local ratings
σ_i	0.5	Threshold of referral generation
ω_i	0.5	Threshold of trust

Table 1. The parameters in the experiments

4.2. Metrics

We now define some useful metrics with which to intuitively capture the results of our experiments.

Definition 3 Suppose $\{W_1, \ldots, W_L\}$ are exactly L witnesses for peer P_j, then the rating distance is defined as

$$|\mathcal{P} - s| \tag{8}$$

where $\mathcal{P}$ is the prediction from the testimonies and s is the rating for the new service from P_j.

Definition 4 The average weight of a witness W_i is

$$\mathcal{W}_i = 1/N \sum_{i=1}^{N} w_i \tag{9}$$

where w_i is the weight of witness W_i from peer P_i's acquaintance model, and N is the number of peers in whose acquaintance model W_i occurs.

Definition 5 The reputation of a peer P_j is defined as:

$$\mathcal{R}_j = 1/N \sum_{i=1}^{N} R(P_i, P_j) \tag{10}$$

where $R(P_i, P_j)$ is the local rating about P_j and N is the number of peers in whose acquaintance model P_j occurs.

4.3. Simple or Exponential Averaging

In the experiments, we assume P_i always queries other peers about the ratings of P_j before it interacts with P_j. The querying peer aggregates the ratings based on the weights it has assigned to the witnesses. The rating distance of each prediction is computed as Equation 8. We average the rating distances of normal agents in $\mathcal{G}_1$ after every 100 round (each peer queries once). Figure 2 illustrates the average rating distances of normal peers, where there are 10% or 30% malicious peers in the system. The experiments show that average rating distances for both cases decrease due to the use of reputation mechanism. [2] Also, peers may use either simple averaging or exponential averaging for their local ratings. We find that averaging techniques make no significant difference if the strategies of peers are fixed, e.g., a peer in $\mathcal{G}_2$ always gives low-quality services and complementary ratings.

However, a reputation mechanism using simple averaging may not be that sensitive to the attacks if the strategies of malicious peers change. In the second simulation, we show if the reputation mechanism is robust against the attacks from peers who may accumulate a high reputation first and then attack other peers in the system. A *malicious* peer P_j who accumulates a high reputation during the first simulation cycle of 2,000 or 20 queries/per peer, behaves cooperatively ($\mathcal{Q} = 0.9$) until it reaches a high reputation value, and then attacks other peers in the system by providing poor services ($\mathcal{Q} = 0.1$). Thus its average reputation begins to drop, ultimately settling at a reputation of 0. A reputation of 0 indicates that P_j is no longer an acquaintance of any peer. In our approach, each peer P_i maintains a blacklist for malicious peers who are swapped out from his acquaintance model. P_j is in the blacklist of the corresponding peer at this time. Figure 3 illustrates the changes of reputation of P_j in the whole process. The results indicate that our reputation mechanisms using either simple averaging or exponential averaging can detect the attacks from strategic peers. The reputation mechanism using exponential averaging is more effective in detecting these attacks, where the reputation of the malicious peer P_j drops quickly from 0.89 to 0.20 during 400 cycles. The malicious peer P_j may also attack the P2P search process through providing wrong referrals. One solution is that every requesting peer may penalize the weight for the credibility of P_j so that they will have a reduced chance of contacting P_j in the future. We will not discuss the details in this paper and defer this enhancement to future work.

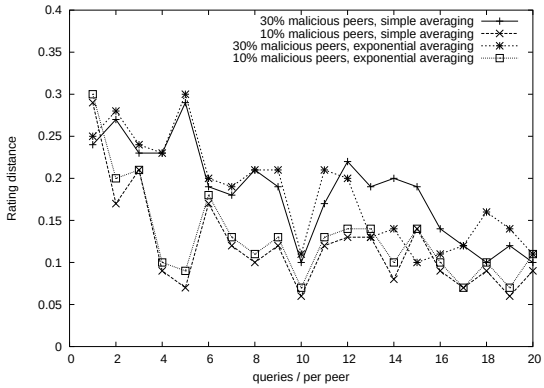

Figure 2. Average rating distances of normal peers in the system

4.4. Credibilities of Independent Peers

One reason that the requesting peer can make better prediction is that it adjusts the weights for witnesses with different credibilities. [3] Therefore, the testimonies from lying witnesses will have less effect on the process of testimony aggregation. Figure 4 shows the change of average weights of witnesses with different credibilities: normal (70%), complementary (10%), exaggerated positive (10%), and exaggerated negative (10%), where peers use simple averaging for their local ratings. We find the weights for witnesses with normal ratings are almost the same, but the weights for witnesses with complementary ratings change a

2 The occasional increases of average rating distances are caused by the process of neighbor selection, which occurs every 5 queries for each peer. Mote that the occasional increases become smaller and smaller in our experiments.

3 Another reason is the process of neighbor selection, which helps peer choose honest or reliable peers as their neighbors.

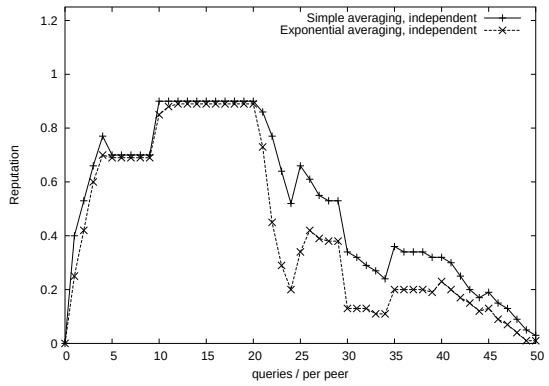

Figure 3. Reputation of an independent peer in group $\mathcal{G}_2$

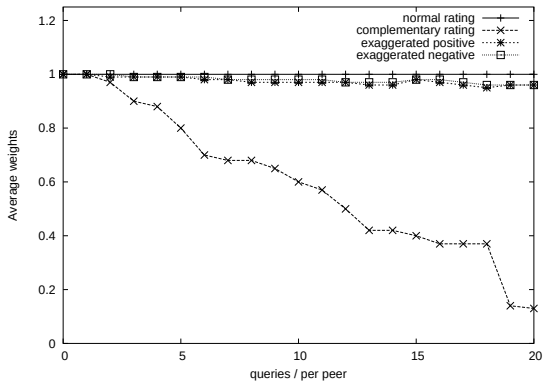

Figure 4. Average weights of witnesses with different credibilities

lot. For the 10 witnesses with complementary ratings, their average weights decrease from 1 to about 0.13 after 2,000 cycles or 20 queries/per peer. Note that, for any peer who gets testimonies from the lying witnesses W_i, its weight w_i from the peer drops much faster than its average weight.

4.5. Effects of Exaggeration Coefficient

The default exaggeration coefficient for witnesses with exaggerated positive or negative ratings is 0.1 in our previous experiments. The present experiment studies the average weights for such witnesses with different exaggeration coefficients. Figure 5 shows the average weights for witnesses with exaggerated negative ratings when exaggeration coefficient α is set to 0.1, 0.2, and 0.3, respectively, where peers use simple averaging for their local ratings. The results indicate that our approach can effectively detect witnesses lying to different degrees. For the 10 witnesses with exaggerated negative ratings, their average weights decrease from 1 to about 0.96, 0.91, and 0.86, respectively, after 2,000 cycles or 20 queries/per peer.

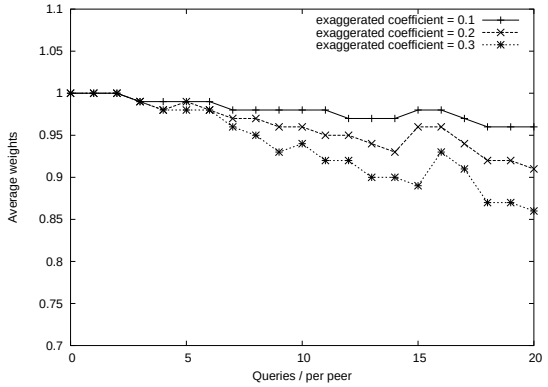

Figure 5. Average weights of witnesses for different exaggeration coefficients

4.6. Attacks of Collusive Peers

Our previous experiments study various attacks from independent peers. In this section we discuss if our reputation mechanism is robust against attacks from collusive peers, where they give good ratings of each other in the group and bad ratings of other peers. Figure 6 illustrates the average weights of 10 independent or collusive peers in group $\mathcal{G}_2$, where peers use simple averaging for their local ratings. We assume the strategies of peers in group $\mathcal{G}_2$ do not change during the experiment. We find the average weights of 10 collusive peers drop slower (measured by the number of queries sent by each peer) because they got positive ratings from peers in the colluding group.

We also study the change of the reputation of collusive peers. We got the similar conclusion as for the weights of collusive peers. In other words, some peers may be cheated more than once by the malicious peers in the colluding group where they cannot identify the lying peers in the system. This often happens when the trusted community has not been created yet in the system. It would be interesting to study how to further minimize the effects of attacks from collusive peers in reputation mechanisms, especially when these collusive peers are not from a clique of IP addresses and traditional network techniques for IP clustering are not helpful.

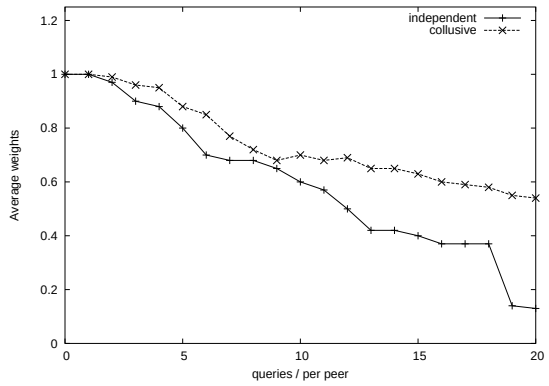

Figure 6. Average weights of independent or collusive peers in group $\mathcal{G}_2$

5. Security Considerations

In this section we discuss some possible attacks of reputation mechanisms [5, 6]. These attacks aim at exploiting the weakness of reputation-based approaches, such as rumor, deception, and pseudonym. Note that [5] and [6] can address all others except rumors.

5.1. Rumors

The problem of rumors is not discussed in other reputation-based approaches. Rumor is different from deception. Rumor happens when the witness returns the ratings not from its direct interaction. These ratings are circulated in the P2P systems and may be used by some malicious peers, e.g., attacking the reputation of normal peers. In our approach of trust graphs the testimonies are based on direct, independent observations, not on communications from others. As a consequence, we are assured that the testimonies can be combined without any risk of double counting of evidence. Double counting of evidence is risky in a distributed system, because it leads to rumors: peers holding opinions about others, just because they heard them from someone.

5.2. Deception

Deception happens when one witness returns multiple ratings or wrong ratings toward a peer. The problem of multiple ratings is relatively simple if the malicious peer encrypt the message with the same key *K*. In this case the requesting peer simply discards all the messages encrypted with key *K*. Sometimes the malicious peer may steal the keys of others. For example, one malicious peer P_m may act as a requesting peer to query other peers multiple times in the system. Once the malicious peer P_m has acquired keys of others, it may send multiple ratings toward peer P_j and encrypt the ratings with different public keys in its local name space. This problem can be partially prevented with IP clustering techniques. The techniques detect if the IPs come from a clique. The requesting peer only takes one from the cluster if a cluster of ratings is detected. [5] gives more details about the technique.

Our approach focuses on the deception from a colluding group where the ratings are from a cluster of malicious peers, but they are not from a clique of IP addresses. Our approach partially prevents this kind of attack with tracking the weights of different witnesses in the colluding group. If one witness is found lying, testimonies from the deceptive witness will have a reduced effect on the aggregated ratings in the future.

5.3. Pseudonyms

In P2P systems, it is relatively easy for peers to disappear and re-enter under a completely different identity with zero or very low cost. The problem often refers to "cheap pseudonym" [9]. Peers can build up a reputation, use it by cheating or attacking others, and then re-enter the systems with a new identity.

Our approach makes it more difficult to change the identities. When a new peer enters into the systems, it has to establish some connections with other peers through key exchanges. The existing peers may accept and add the new peer to the local name space, or simply decline the new peer's request. The process may involve human interactions and could be expensive. Also, a high reputation is hard to accumulate but is easy to be destroyed. The information of attacks from these peers will be quickly spread through the system. The diameter of practical P2P systems can be quite small [3]. This reduces the possibility of attacking others with the same identity. Even the peer drops its identity and comes up with a new identity. The new identity with low reputation will dramatically reduce the possibility of being chosen for future interaction [5], due to the possible trusted community formation.

6. Conclusion

In this paper we propose a robust and efficient reputation mechanism for large-scale P2P systems, in which a peer combines testimonies from several witnesses to determine the trustworthiness of another peer. We focus on how to effectively detect deception in the process of ratings propagation and aggregation. Our approach improves the security level of P2P systems without the need of trusted third parties and alleviates some of the security problems in P2P systems, e.g., identifying and blocking some malicious peers.

In future work, we plan to study the possible trusted community formation in dynamic P2P systems with certain departure rates and arriving rates. We also want to study reputation mechanisms in P2P systems with other topologies, e.g., power-law or scale-free networks, where peers have different numbers of neighbors and acquaintances. Our goal is to develop a robust distributed trust model for large and dynamic P2P systems and help peers manage the risk that is involved with unknown parties in large-scale P2P systems.

Acknowledgements

This research was supported by the National Science Foundation under grant No. ITR-0081742, the AFOSR under grant No. F49620-01-1-0542, and by AFRL/MNK grant No. F08630-03-1-0005. We are indebted to the anonymous reviewers for their helpful comments.

References

[1] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 164–173, 1996.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference (WWW7)*, pages 107–117, 1998.

[3] S. Capkun, L. Buttyan, and J.-P. Hubaux. Small worlds in security systems: an analysis of the PGP certificate graph. In *Proceedings of the ACM Workshop on New Security Paradigms*, 2002.

[4] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making gnutella-like P2P systems scalable. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 407–418, 2003.

[5] F. Cornelli, E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati. Choosing reputable servents in a P2P network. In *Proceedings of the Eleventh International World Wide Web Conference*, pages 376–386, 2002.

[6] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *Proceedings of the Ninth ACM Conference on Computer and Communications Security*, pages 207–216, 2002.

[7] J. R. Douceur. The sybil attack. In *Proceedings of First International Workshop on Peer-to-Peer Systems*, 2002.

[8] C. M. Ellison. Establishing identity without certificate authorities. In *Proceedings of the 6th USENIX Security Symposium*, pages 67–76, 1996.

[9] E. J. Friedman and P. Resnick. The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy*, 10(2):173–199, 2001.

[10] N. R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multiagent Systems*, 1(1):275–306, 1998.

[11] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In *Proceedings of the Twelfth International World Wide Web Conference*, pages 640–651, 2003.

[12] H. Kautz, B. Selman, and A. Milewski. Agent amplified communication. In *Proceedings of the National Conference on Artificial Intelligence*, pages 3–9, 1996.

[13] N. Li, B. N. Grosof, and J. Feigenbaum. Delegation logic: A logic-based approach to distributed authorization. *ACM Transactions on Information and System Security (TISSEC)*, 6(1):128–171, 2003.

[14] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[15] S. Marti and H. Garcia-Molina. Limited reputation sharing in P2P systems. In *Proceedings of the ACM Conference on Electronic Commerce*, 2004. to appear.

[16] M. Paolucci, K. Sycara, T. Nishimura, and N. Srinivasan. Using DAML-S for P2P discovery. In *Proceedings of the First International Conference on Web Services*, pages 203–207, 2003.

[17] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proceedings of ACM SIGCOMM*, pages 161–172, 2001.

[18] M. K. Reiter and S. G. Stubblebine. Toward acceptable metrics of authentication. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 10–20, 1997.

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

[20] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. In *Proceedings of the Second International Semantic Web Conference*, pages 351–368, 2003.

[21] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of ACM SIGCOMM*, pages 149–160, 2001.

[22] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, June 1998.

[23] B. Yu and M. P. Singh. An evidential model of distributed reputation management. In *Proceedings of First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 294–301, 2002.

[24] B. Yu and M. P. Singh. Detecting deception in reputation management. In *Proceedings of Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 73–80, 2003.

[25] B. Yu, M. Venkatraman, and M. P. Singh. An adaptive social network for information access: Theoretical and experimental results. *Applied Artificial Intelligence*, 17(1):21–38, 2003.

[26] T. Yu, M. Winslett, and K. E. Seamons. Interoperable strategies in automated trust negotiation. In *Proceedings of the 8th ACM Conference on Computer and Communications Security*, pages 146–155, 2001.