GroupRep: A Robust Group-based Reputation System for Peer-to-Peer Networks

Ye Tian, Di Wu and Kam-Wing Ng
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{ytian, dwu, kwng}@cse.cuhk.edu.hk

Abstract

Reputation systems, which allow peers to collaborate on estimating the trustworthiness of other peers, are considered as one of the promising directions against malicious behaviors in large scale Peer-to-Peer networks. However, a weakly designed reputation system is vulnerable to attacks such as unfair rating and collusion. This paper presents GroupRep - a novel group-based reputation system for Peer-to-Peer networks. In the GroupRep model, peers are allowed to form social groups, in which member peers will collaborate against attacks on the reputation system. For management of the group, a set of robust and light-weighted protocols are proposed based on distributed voting. Experiments show that the GroupRep model is effective and inexpensive, and the system is robust enough even under the situation that the majority of the peers are malicious and they cooperate deliberately to subvert the system.

1. Introduction

Peer-to-Peer (P2P) technology, which enables users to contribute and share their resources, has become very popular in recent years. However, the dark side of this novel technology is also emerging with its success. It is reported that 45% of the files found by popular keyword searches in Kazaa [2], which is a popular file-sharing system, were infected with virus [1]; and more and more malicious contents such as worms or even spywares are found spreading with the facilities provided by P2P technology. Since the P2P overlay is considered as the next generation platform for deploying some critical services such as electronic commerce, security as well as privacy issues, have become one of the major concerns in P2P research.

Usually different attack schemes will be used against different applications by a malicious user, however, regardless of the material application background, it is essential for the system to identify the malicious peer and warn its existence to others for the purpose of security. Trust and reputation systems, in which peers collaborate to safeguard each other by sharing their historical experiences, are considered as one of the promising directions against malicious behaviors in P2P networks. In recent years, a number of systems [3] [5] [6] [7] [11] [12] have been proposed in this area. However, like any systems and services deployed upon the P2P overlay, the trust and reputation system could also be manipulated because of its decentralized nature. For example, in an unfair rating attack, peers will issue dishonest feedbacks; and in a collusion attack, colluding peers could cooperate by spreading misleading trust information to exploit the system.

How to filter out useless or misleading recommendations (or recommenders) is the essential problem for building a robust reputation system. Some recent researches [6] [7] have proposed solutions for the problem, however, they are either not robust enough (such as associating the recommender’s trust with the recommendations it has issued) or require very strong assumptions (such as the accessibility of global trust information).

In this paper, we present GroupRep – a novel group-based reputation system for P2P networks. In the GroupRep model, peers can form a self-organized social group, denoted as TrustGrp. Peers in the same TrustGrp will share their recommendations proactively and monitor each other’s trustworthiness both in providing service and in issuing recommendations. The first unique contribution in GroupRep is: unlike the system in [6], peers are not required to access any global trust information, but only need to evaluate a recommendation based on local trust information available within a TrustGrp. The second contribution for GroupRep is: for management of the TrustGrp, we propose a set of robust and light-weighted TrustGrp management protocols.
based on distributed voting, in which a peer’s membership of a TrustGrp is updated periodically according to its behavior. Finally, we show via simulation that the GroupRep model is robust enough under the attacks of unfair rating combined with collusion. The protocol runs at a small communication overhead. And more surprisingly, even under the situation where a majority of the peers are being malicious and cooperating deliberately to subvert the system, peers in a TrustGrp can still filter out malicious peers effectively.

The remainder of the paper is organized as follows. Related work is discussed in Section 2. In Section 3, we present the GroupRep model in details. For Section 4, simulation settings and experimental results are given for performance evaluation. Finally, Section 5 concludes the paper and discusses the future work.

2 Related Work

Among the enormous trust and reputation schemes proposed for P2P networks, XRep [3] is regarded as the first practical reputation system, which is deployed on Gnutella [4]. In XRep, to evaluate a target peer’s trustworthiness, a requesting peer will use a Poll message to initiate a vote. When a peer receives a Poll message, it will return its opinion with a PollReply message, and then the peer initiating the voting will have a view on the target peer’s trust level from the opinions it has gathered. However, the opinion-based recommendation has limitations since peers could give dishonest feedbacks without any constraints.

Aberer and Despotovic [5] considered the authenticity of the recommendations and proposed a decentralized trust and reputation system. In their model, a negative feedback is kept as an evidential recommendation called a complaint and is stored on a structured overlay called P-Grid [8]. The evidential recommendation mechanism works well against the attack from peers issuing fake recommendations without real interactions, however, it cannot detect and stop peers from lying in recommendations based on interactions that have really happened.

Some recent models focus on mechanisms against possible attacks such as unfair rating and collusion. For example, in PeerTrust [6], each peer is associated with a reliability measurement about the quality of recommendations it has issued. Two algorithms are proposed for eliciting the reliability: the Trust Value Measure (TVM) algorithm and the Personalized Similarity Measure (PSM) algorithm. The former uses a function of the trust value of a peer as its credibility factor while the latter uses similarity to rate the credibility of another peer by comparing all the recommendations the two peers have issued. The authors compared the two solutions and concluded that PSM outperforms TVM.

Both in Aberer and Despotovic’s model and in PeerTrust, peers are assumed to have the ability of accessing global information to form a global trust and reliability view. Usually, a DHT overlay such as P-Grid is assumed to provide this functionality. However, DHT itself is prone to numerous possible attacks, such as those mentioned in [9]. And for a large system, accessing and evaluating the global trust information is a laborious job.

On the other hand, [10] points out that if peers are organized with social structures, the robustness of the reputation system will be improved. However, to our knowledge, very few systems support a social structure among peers. In [11], peers with different service categories can form a group and use the group’s reputation as a virtual currency, however, the purpose of the group is not to improve the robustness and the system cannot work well under the collusion attack. Another system called Buddy system [12] attempts to improve the robustness of the reputation system by allowing two peers to form a “buddy” relationship, however, this system can only support bilateral groups which reduce its effectiveness greatly.

3 The GroupRep System for Peer-to-Peer Networks

In this section, we will describe the design of GroupRep in details. Our reputation system is purely distributed among all the peers; each peer runs an instance of the system. There are two components for each GroupRep instance: the trust and reputation management component and the trust group management component. The former is responsible for organizing and publishing the trust information while the latter is used for building and maintaining TrustGrp based on distributed voting.

3.1 Trust and Reputation Management

The raw data for the GroupRep system is evidential recommendations. It is assumed that for any interaction, only two peers are involved, the service provider (SP) peer and the service consumer (SC) peer. We also assume that a self certification scheme is used for identification of peers, in which each peer is associated with a public/secret key pair. When two peers have finished an interaction, an evidential recommendation on the trustworthiness of the SP peer in this interaction will be generated by the SC peer. An evidential recommendation is in the form of \( RC^j_i = \{(p_j.ID, p_i.ID, tstamp)_{sk_j, ts}, \) \( sk_i \). Here \( p_j.ID \) and \( p_i.ID \) are identities of the SP peer and SC peer respectively, \( tstamp \) is the time stamp of the interaction and \( ts \in [-1, 1] \) is the trust score given by the SC peer \( p_i \) according to its perception on the quality of service provided by SP, here we specify the positive trust scores for good service and the negative ones for malicious interactions. The
For example, the weighted average could be calculated as the direct trust entry periodically, which means we re-
commendations issued by peer
recommendations not issued by itself, however, we do not
Once a recommendation is issued, it is used as a refer-
ence for future potential SC peers when they are making
decisions. Concretely, a potential SC peer will launch a
query process as in XREP on the network for recom-
dendations on a particular SP peer if it lacks relative trust
information. On receiving such a query, a peer will lookup its
trust database and choose to answer the query with appro-
priate recommendations; of course, it could also ignore the
query. After the query process, all the recommendations
received by the querying peer are cached on the peer whether
or not there is any subsequent interaction. The querying peer
could answer future queries on this particular SP peer with
these caches. However, when a recommendation is issued by a peer in a TrustGrp, it will be proactively broadcasted by the issuing peer to all the members of the Trust-
Grp. In this way, recommendations in TrustGrps are distri-
buted more efficiently.

Any peer in our system will maintain a direct trust ta-
ble for all the other peers it has interactions with directly.
Suppose a peer \( p_i \) has some interactions with peer \( p_j \), the entry for \( p_j \) is denoted as \( T_{s_j} = n, ts, RC\_LIST \), where \( n \) and \( ts \) are the number of recommendations and av-
ged direct trust score of these recommendations, and \( RC\_LIST \) is an index in which these recommendations are kept. In GroupRep, we divide the time into rounds, and update the direct trust entry periodically, which means we re-
calculate the \( T_{s_j}, n \) and \( T_{s_j}, ts \) by considering new re-
commendations issued by peer \( p_i \) since the last update. For example, the weighted average could be calculated as

\[
T_{s_j}, ts = \sum_{RC_i \in T_{s_j}, RC\_LIST} w_i \cdot RC_i, ts
\]

\[
T_{s_j}, n = \sum_{RC_i \in T_{s_j}, RC\_LIST} w_i \cdot n
\]

For each \( RC_i \), the weight \( w_i \) decays with time as \( w_i = \alpha^{t - tstamp_i} \), where \( t \) is the current time and \( \alpha \) is the dec-
ay factor in the range of \((0, 1)\).

Similarly, a peer will derive a reputation from the re-
nommendations not issued by itself, however, we do not al-
low a peer to have reputation entries for peers in the
same TrustGrp. On peer \( p_i \)'s reputation table, an entry for peer \( p_k \) outside \( p_i \)'s TrustGrp is denoted as \( R_{sk} = n, rs, RC\_LIST \), with \( n \) and \( rs \) having the same mean-
ings as in the direct trust table. We also update the reputa-
tion table periodically by recalculating the weighted av-
erage of recommendations, however, the reliability of the
recommendation is represented by the weight since these
recommendations are not issued by the calculating peer it-
self. Actually, if the recommender peer \( p_j \) is not a group
mate of peer \( p_i \), we have weight \( w_l \) as a scale of peer \( p_j \)'s
direct trust \( T_{s_j}, ts \), otherwise, we let \( w_l \) to be the peer \( p_j \)'s
recommendation trust \( R_{t_j}, rt \), which will be described later.
Of course, \( w_l \) should also decay with time.

Peers could form a belief on the trustworthiness of a par-
ticular peer outside its group by combining the direct trust
and reputation. This belief is updated for each round, and
we use the belief for evaluating the reliability of recommen-
dations received in the next round.

As just mentioned, if the recommender peer \( p_j \) is a group
mate of peer \( p_i \), it will be associated with a recommendation
trust entry \( R_{t_j} \) for the quality of the recommendations it has
issued. We denote \( R_{t_j} \) as \( R_{t_j} = n, rts, RC\_LIST \), here the \( n \) and \( rts \) are the number of recommendations and the averaged recommendation trust score for the quality of all the recommendations \( p_j \) has issued, and \( RC\_LIST \) keeps an index of these recommendations. For each recom-
endation, an evaluation function \( f \) is used for comparing the
belief on the target peer the peer \( p_i \) holds and the trust
score given by \( p_j \) in the recommendation. For example, for
the recommendation \( RC^k \_j \), the recommendation trust score is
\( f(RC^k \_j, ts, bs_k) \), where \( bs_k \) is the belief on peer \( p_k \) held by peer \( p_i \). Here we do not further specify the eval-
uation function, any function which can express the differ-
ence could be used here, for example, it could be as simple
as a scale of \( |RC^k \_j, ts - bs_k| \), which is used in our simu-
lation. The \( R_{t_j}, rts \) is calculated as the weighted average recommendation trust score for all the recommendations in \( RC\_LIST \) and each decays with time.

### 3.2 Trust Group Management

In GroupRep, peers need to form TrustGrps to collabor-
orate in exchanging recommendations. Technically, each
TrustGrp group is initialized by a peer called the founder,
and other peers could choose to join in the TrustGrp by
applying to the founder peer. For a particular TrustGrp,
the founder peer will decide on a group public/secret key.
Each member peer of this particular TrustGrp will keep a
group member certificate (GMC) which states its member-
ship, for example, peer \( p_i \)'s GMC could be in the form
\( (p_i.ID, p_j.ID)_{sk_g} \), where \( p_j.ID \) is the group founder
peer’s ID, and the GMC message is signed with \( sk_g \), which
is the secret key of this TrustGrp. Because of the dynamics
of the P2P networks and of the peer’s behavior, in a Trust-
Grp, a member peer’s GMC should be updated periodically
to adapt to these changes.

We use the Threshold RSA [13] to issue and manage the
GMC for the TrustGrp. In a \( K/N \) Threshold RSA, the
secret key is divided into \( N \) secret shares. For a message, if it
is signed with \( K \) secret shares, a valid signature of this mes-
sage signed by the original secret key could be constructed
from these \( K \) partial signatures.

In our TrustGrp management protocols, the basic proce-
dure is a distributed voting based on Threshold RSA. For a voting, two sets of peers are required, one is the candidate set of peers which will be voted on for their membership, denoted as candidate – set, and another is the voter peers which have the voting right, denoted as voter – set. During the voting procedure, each candidate peer will generate a GMC message and contact each voter for its partial signature. For each voter peer with a secret share of the group’s secret key, on receiving such a request, it will choose whether or not to sign the GMC message according to some trust-related criteria. After all the voter peers have made their decisions, those candidate peers which have successfully gathered K partial signatures could construct a valid GMC and declare its membership to other peers.

Based on the distributed voting procedure, we develop three protocols for the management of the TrustGrp, which are for the bootstrap phase, the eviction phase and the admission phase. The bootstrap protocol is applied when a TrustGrp is initialized by a founder peer p0, and there is a set of peers \{p1, . . . , pn\} which apply to join in. The protocol set is candidate = set = voter – set = {p1, . . . , pn}, and a distributed voting is started allowing the applying peers to vote on each other. For the eviction protocol, it is executed periodically to renew the membership. At the eviction phase, a founder peer p0 will decide a new public/secret key pair for the TrustGrp, and distribute the secret shares to all the former member peers. Similarly, the protocol set is candidate = set = voter – set = \{p1, . . . , pn\}, which is the set of the former member peers and a distributed voting is started for them to obtain their new GMCs. For the admission protocol, it is triggered when there is a set of peers \{p_1', . . . , p_n'\} which apply to join in the TrustGrp. In the admission phase, we let voter – set = \{p_1, . . . , p_n\}, which is the set of member peers, and let candidate – set = \{p_1', . . . , p_n'\}, which are the applying peers. By distributed voting, the member peers could vote for new peers to join in.

For the three phases of the TrustGrp management, a peer will make its voting decisions by referring to its direct trust table, reputation table and recommendation trust table. Generally, a member peer will wish the other members to be good both at providing service and issuing informative recommendations, so, for the eviction protocol, peers will not vote for those peers which have failed in either of the two. However, for the bootstrap and admission protocols, since the voter peers do not have recommendation trust table entries for those peers outside the TrustGrp, they will simply vote for peers providing good services. Actually, a peer will use some personalized thresholds when making its decisions. This simple voting principle is robust, since we require that a member peer should provide good services and reliable recommendations at the same time, so, for those colluding peers which provide good service but issue misleading recommendations, they may have chances to get admitted in, however, it is hard for them to stay in a TrustGrp for a number of continuous rounds.

4 Evaluation

We have performed three experiments to evaluate the performance of the GroupRep model under the attacks of unfair rating and collusion. In the first experiment, we have compared the effectiveness of the GroupRep system with a TVM-like model and a PSM-like model [6]. For the second experiment, we have studied the performance of the system under different percentages of malicious peers in the P2P community. And for the third experiment, we have investigated the communication overhead and the scalability of the GroupRep system and compared it with TVM-like and PSM-like reputation schemes.

We simulate a P2P community of 500 peers in our experiments. The peers are categorized into three classes, which are altruistic, selfish and malicious. For any interaction between two peers, the quality of service provided by the SP peer is measured with a score in the range of [−1, +1]. We regulate the behaviors of peers as follows. For an altruistic peer, it always provides a high-quality service when requested, and reports honest recommendations. However, for a selfish peer, it only provides a low-quality service, and issues a random recommendation for the SP peer which it has interactions with, thus launching an unfair rating attack. For the malicious peers, we allow them to attack the reputation system by colluding. Actually, the malicious peers are divided into two sets, one set of peers are action peers, which will behave maliciously in each interaction by providing bad contents with the quality of −1; for the other set of peers called front peers, they will provide services as the altruistic peers, but issue the highest positive recommendations for the action peers.

For the first experiment, we set the percentage of altruistic peers as 40%, selfish peers as 50%, and malicious peers as 10%. For simplicity, there is only one TrustGrp in the network, which is composed of about 20 peers. We control the size of the TrustGrp by changing the parameter K and N of the distributed voting procedure. In Figure 1, we compared the GroupRep system with the TVM-like model and the PSM-like model. Four types of peers were studied in the experiment: non-grouped peers in GroupRep, grouped peers in GroupRep and peers in the TVM-like and PSM-like models. The success rate is computed as the sum of quality of services obtained by a particular type of peers divided by the total number of interactions. From Figure 1(a), we can see that the PSM-like model achieves the best success rate since it makes used of the global information, and the GroupRep system outperforms the TVM-like model for the grouped peers. We also consider the number of ma-
licious interactions conducted in each round, it is obvious from Figure 1(b) that for both the PSM-like model and the grouped peers in the GroupRep system, malicious peers are effectively identified even they launch a collusion attack.

In the second experiment, we deteriorated the community by changing the malicious peer percentage from 0% to 80% in the GroupRep model. Figure 2 shows the success rate and number of malicious interactions conducted at the 200th round for peers in and out of the TrustGrp. It is observed that although the success rates of both types of peers are decreasing with increase of the malicious percentage (Figure 2(a)), the grouped peers could avoid having interactions with malicious peers even under the extreme condition of 80% malicious peers (Figure 2(b)).

For the third experiment, we scaled the size of the network from 50 peers to 500 peers, and studied the communication overhead for a single peer under different reputation schemes. We calculate the overhead as the sum of the number of recommendations transferred and the number of messages for protocol maintenance; the size of the TrustGrp is kept as $10 \log n$ for GroupRep, where $n$ is the network size. The result is plotted in Figure 3. It is observed that both the TVM-like model and the PSM-like model scale linearly with the size of the network; however, for the GroupRep system, the scalability is logarithmic. This could be explained that for a peer in a TrustGrp, it only needs to communicate with its group members, and manage the recommendations issued by its member peers; but for a peer playing the TVM algorithm or the PSM algorithm, it must access and compare the recommendations issued by all the peers in the system. We conclude when the network size is small, both the PSM and the TVM models have smaller communication overhead than the GroupRep system, in which some traffics are caused by the TrustGrp management protocols, however, their overheads increase rapidly with the network size. On the other hand, GroupRep achieves a better scalability in the communication overhead for larger scaled networks.

Figure 3. Protocol overhead of GroupRep, compared with the TVM-like reputation model and the PSM-like reputation model.

5 Conclusion and Future Work

We have presented GroupRep - a novel group-based reputation system for P2P networks in this paper. By allowing peers organized as TrustGrps, the system provides robustness against attacks such as unfair rating and collusion. We have also proposed a set of robust and lighted-weighed TrustGrp management protocols based on distributed voting. Some initial experimental results are given, showing that GroupRep is effective under these attacks even under some extreme conditions, while is inexpensive for communication overhead. For the future work, one of the ongoing
researches is to study the incentives in the GroupRep system, some economic model such as free market would be worth exploring.

6 Acknowledgments

The work described in this paper was partially supported by the following grant: RGC Research Grant Competitive Bids (Project ID: 2150414; Reference: 4220/04E).

References


