BGET: A Blockchain-Based Grouping-EigenTrust Reputation Management Approach for P2P Networks

Yang Peng[1(✉)], Jie Huang[1,2], Sirui Zhou[1], Zixuan Ju[1], Xiaowen Wang[1], and Peihao Li[1]

[1] School of Cyber Science and Engineering, Southeast University, NanJing 211189, Jiangsu, China
{seu_py,jhuang,220224873,220224737,xiaowenwang,lipeihao}@seu.edu.cn
[2] Science and Technology on Communication Networks Laboratory, Shijiazhuang 050000, Hebei, China
http://www.springer.com/gp/computer-science/lncs

Abstract. Trust, as an effective way to reduce complexity and risks of systems, is experiencing various challenges in distinguishing reliable partners of distribution environment like Peer to Peer (P2P) networks. As one of the most known and successful reputation management systems, the EigenTrust reputation management system has been widely used. However, this kind of system uses centralized reputation calculation strategy, which relies heavily on the mechanisms of global ranking and pre-trusted peers. It causes high-reputation peers to center around pre-trusted peers and uncontrolled spread of inauthentic downloads, as a consequence, other low-reputation peers will be marginalized despite potentially they could be honest. To deal with these problems, we put forward a blockchain-based grouping-EigenTrust (BGET) reputation management approach. BGET uses grouping-ET algorithm to manage reputations of different peers, which utilizes uniform grouping strategy and intragroup random walk strategy to divide peers into different groups to guarantee the uniform distribution of high reputation peers and limit the spread of inauthentic downloads. Moreover, BGET provides reliable verification services based on blockchain, which can improve the credibility and quality of transactions. Through simulations, we proved that BGET has good extensibility and robustness. BGET can effectively maintain higher success rate of tasks even there are many malicious peers in the network.

Keywords: P2P Networks · Reputation Management · EigenTrust · Blockchain

1 Introduction

P2P networks are designed to improve network resilience, in which members are peer to peer and no longer subject to or subordinate to each other. Up to today,

H. Gao et al. (Eds.): CollaborateCom 2023, LNICST 561, pp. 225–242, 2024.
https://doi.org/10.1007/978-3-031-54521-4_13

P2P networks are used for a wide range of applications, from resources sharing [17] and social networking [11,13] to Internet of Things [19,20], electricity trading [27] and governance [8]. They offer advantages over traditional networks, including increased resilience, transparency, and security. However, how to effectively measure trust relationships among unacquainted peers in these networks is a longstanding challenge.

To ensure the security of P2P networks, a reliable reputation system is essential, which can identify peers who are honest and reputable, while also detect those who are deceitful, malicious, or self-serving. Reputation is a global perception of an entity's behavior based on the trust that other entities have established [18]. Therefore, many reputation systems have been suggested, such as EigenTrust (ET) [7], PeerTrust [24] and PowerTrust [29], etc. However, these systems need to rank the reputation values of their members to distinguish honest peers from malicious ones. This will lead to some fatal problems that peers in the system will naturally cluster around the peers with high reputation and the effects of low ones will be marginalized. Even ET, one of the most popular and used reputation system, can't get away with these.

ET was first proposed for P2P file-downloading systems to help peers acquire resources from trusted peers [7], which is achieved by calculating a global reputation value, consists of a local trust value determined by peer's previous behavior and a propagation-based recommendation trust value, for each peer and sorting them. ET has two essential mechanisms, a group of pre-trusted peers and global ranking, which are utilized to help peers choose who to interact with. Obviously, the pre-trusted peers have more opportunities to earn high reputations, since they have higher initial reputation values to make them trustworthy. However, this also prompts other peers, who expect to gain high reputation, to gather around and connected with them. In this condition, the pre-trusted peers could become a wonderful medium to spread inauthentic downloads. If they accept downloads from malicious peers accidentally, they will help spread these inauthentic files to the entire network more efficiently, since there are many peers connected to them. Additionally, feedback and evaluation in ET come from the subjective level of the service requester, which is very beneficial for malicious peers to engage in fraudulent activities. Because honest peers always tell the truth, while malicious ones can intentionally lower their ratings for services provided by honest peers, in order to lower their reputation rankings (i.e. bad-mouthing attack [6]). Although ET performance well in many scenes, it causes peers to converge around the pre-trusted peers and high-ranking peers, and the peers that are 'far away' from the them will be ranked very low despite potentially being honest [4]. To sum up, the primary limitation of ET is the aggregation problem caused by pre-trusted peers and global ranking.

Contributions: we propose the BGET approach for P2P networks to maintain the uniform distribution of high reputation peers and provide a fair collaborative environment for peers, where there is no longer the concentration of high-reputation peers and the marginalization of low-reputation peers. Among the main contributions of this paper are:

- We characterize the negative implications of ET: pre-trusted peers and global ranking, as shown in Sect. 3. Peers that are pre-trusted have higher reputation values and are easier to accumulate reputation values, which also make them more likely to cause uncontrolled spread of inauthentic downloads. If pre-trusted peers receive inauthentic downloads carelessly, then the inauthentic downloads will be spread to the whole network faster. In the long run, the whole network will control by minority peers with high reputation and the influence of low reputation peers on the entire network will decrease and eventually be marginalized. To tackle these problems we propose the blockchain-based grouping-EigenTrust (BGET) reputation management approach for P2P networks.
- To avoid the aggregation of high-reputation peers and limit the spread of inauthentic downloads among pre-trusted peers, we put a uniform grouping strategy into ET, and propose the grouping-ET algorithm to manage the reputation of peers. The grouping-ET algorithm utilizes uniform grouping strategy to divide peers in the network to a uniform state and uses intragroup random strategy to suggest peers select interactive objects that from the same group as themselves in a random way. By doing so, we can provide a more fair trading condition for honest peers by limiting the opportunities of malicious ones to manipulate transactions.
- The transitivity of recommendation trust is a critical process of reputation convergence, and is also a good opportunity for malicious peers to manipulate. To guarantee the reliability of recommendation trust, we introduce blockchain technology into BGET to provide verification services, which can verify the identity and evidences of trust of both parties involved in the transactions.
- We design a series of simulations to test the credibility and performance of BGET. The results show that BGET can achieve better transaction success rates than original ET, and also exhibit better robustness in the face of attacks.

The rest of the paper is organized as follows: Sect. 2 summarizes previous works on reputation management system, especially on ET and blockchain based methods. Section 3 analyzes ET algorithm and its limitations. Section 4 describes our BGET approach to address the problems in exiting reputation management systems. The evaluation process and result discussion are presented in Sect. 5. Finally, a conclusion with a summary is provided in Sect. 6.

2 Related Work

Reputation management methods are designed to collect and analyze data about the reputation of individuals, organizations, and products on various online platforms [5]. Reputation management methods are not only widely used in P2P networks, but also widely used in various scenarios such as social networks [28], internet of things [9], internet of vehicles [23], and so on. With the deepening

of the research, many reputation systems have been put forward. Aberer and Despotovic [1] presented a reputation-based trust management approach, which can be implemented in P2P environment and scales well for very large numbers of participants. In [21], Credence, an innovative reputation system based on objects, is introduced as a solution to evaluate the credibility of online content and combat content pollution by securely collecting and managing endorsements from trustworthy peers. For distribution environment, Xue et al. [25] design a robust and distributing reputation system, DHTrust, which is modeled by DHT (Distributed Hash Table) trust overlay network (DHTON). Can and Bhargava [3] propose a Self-ORganizing Trust model (SORT) that aims to decrease malicious activity in a P2P system by establishing trust relations among peers in their proximity.

ET Algorithm is a famous trust-based reputation management approach, which uses the principle of eigenvector centrality to compute the trustworthiness of each peer. With the further research on ET, many variants and extensions have been proposed to address specific challenges and limitations of the original ET algorithm. Kurdi et al. in [12] propose a HonestPeer algorithm to dynamically select the most reputable peers based on the quality of the shared files, instead of relying solely on a static group of pre-trusted peers. AlhussainK et al. [2] introduce the EERP approach to track, identify and isolate malicious peers by analyzing the logs of peer interactions. Besides, many scholars have studied personalizing ET, and propose a series of methods. Chiluka et al. [4] introduce a reputation system called PETS (Personalized ET using Social network), Lin et al. [15] propose the Personalized ET reputation system, Li et al. [14] propose PersonalTrust pre-trust reputation management model. These researches have studied the pre-trusted mechanism of ET and found some ways to choose the pre-trusted peer based on their own tastes, and can detect various types of malicious behaviors that were not detected by the original ET algorithm. Although these methods are more flexible than ET, the aggregation problem caused by pre-trusted peers still exist. Moreover, many of them overlook the global ranking mechanism, which can also lead to similar problems.

In recent years, blockchain is widely used in internet of things (IoT) to manage the trust relationship of members. Yang et al. [26] propose a decentralized trust system for vehicular networks. Wu and Liang [22] utilize the mobile edge nodes to calculate the trustworthiness of sensor nodes and put forward the blockchain-based trust management mechanism (BBTM) approach. Kouicem et al. [10] design a hierarchical and scalable blockchain-based trust management protocol, which achieve the mobility support in decentralized IoT systems. In order to comprehensively study the development status of blockchain-based trust management systems (BC-TMSs), Liu et al. [16] conduct a serious survey on the current state of the arts, in which introduces the recent advances, open issues, and future research directions toward realizing reliable and sound BC-TMSs for IoT. The decentralized, tamper resistance, and traceable features of blockchain are very suitable for the reputation management requirements in P2P networks,

which can overcome identity impersonation and fraud issues between entities that are unfamiliar.

3 Analysis of ET

3.1 Algorithmic Overview

Considering a file-downloading scene in P2P networks, there are m peers, including a pre-trusted group, P, which is known to be trustworthy. The initial reputation of pre-trusted peers is setting to $\frac{1}{|P|}$.

Assuming there is a peer i, who downloaded files from peer j. Then, there will be a local trust value, S_{ij}, to represent the total number of authentic downloads from peer i to peer j. To avoid malicious peers giving other malicious peers arbitrarily high local trust values, S_{ij} should be normalized as $c_{ij} = \frac{\max(s_{ij},0)}{\sum_j \max(s_{ij},0)}$, representing how much peer i trust peer j based on their historical transaction records. This formula ensures that all values of c_{ij} are between 0 and 1. Up to this point, peer i have completed the trust evaluation of peer j.

To expand its knowledge, peer i need to study the viewpoints of other peers in the network. A feasible approach is to access knowledge from its neighbors and the neighbors of its neighbors. Then, peer i will get a aggregation reputation vector:

$$\overrightarrow{t_i} = (C^\top)^n \overrightarrow{c_i} \tag{1}$$

after n iterations.

In real applications, there will be a group of malicious peers who know each other, and they will give each other high trust values, but give all others low trust values. This issue is addressed by taking

$$\overrightarrow{t}^{(k+1)} = (1-\alpha) C^\top \overrightarrow{t}^{(k)} + \alpha \overrightarrow{p} \tag{2}$$

where α is a constant less than 1.

3.2 Security Analysis

ET has two typical aspects: pre-trusted peers and global ranking, which make it is different from other trust management methods. On the contrary, these two mechanisms also bring performance limitations to it.

The pre-trusted mechanism is designed to avoid the manipulation of malicious peers. Firstly, when there are malicious individuals in the networks, the convergence rate of $t = (C^\top)^n \widetilde{p}$ is faster than $t = (C^\top)^n \widetilde{e}$, where $\widetilde{e}$ represents the previous trust value. Secondly, when there are slothful individuals, which means they don't trust anyone and don't download files from anyone they couldn't trust, then we could define $c_{ij} = p_j$. In other word, if peer i couldn't trust anyone, then i could choose the ones that are pre-trusted. Finally, when there are many malicious peers want to manipulate the process of reputation evaluation,

the pre-trusted peers can relieve this situation. ET suggests every peer to choose the pre-trusted peers with a certain probability α, as shown in Eq. (2). However, the peers close to the pre-trusted ones are more likely to be selected by other peers. So, they can earn more reputation than the peers who are far away from the pre-trusted ones. Chiluka et al. [4] have proved this observation in their researches. They found the peers closer to the pre-trusted ones have higher reputation rankings, while those farther away will be marginalized. Furthermore, if a pre-trusted peer downloads an inauthentic file from a malicious peer carelessly, the wandering mechanism would allow the file to be easily accepted by other peers, leading to a chain of inauthentic downloads [12]. As a consequence, we think the pre-trusted mechanism of ET will lead to the aggregation problem of high-reputation peers, which deviates from the original intention of decentralized design in P2P networks and also lead to uncontrolled spread of inauthentic downloads.

Global ranking is an intuitive way to reflect the trust level of peers in a community. In principle, the higher the ranking, the more trustworthy the peer is. However, this open approach also provides more opportunities for malicious peers to take advantage of. When there are many malicious individuals in the networks, they will improve their reputation by providing authentic downloads to high-reputation ones, while providing inauthentic downloads to low-reputation ones. Besides, they can give high scores to other malicious peers who collude with them, and give honest peers low scores to weaken their influence on the network. In the long run, the negative impact will spread throughout the entire network.

Although the pre-trusted peers and global ranking mechanisms bring the ET good performance to distinguish honest peers from malicious ones and maintain a stable reputation network when there are few malicious peers, the performance will be poor when the number of malicious peers in the network is large. So, how to mitigate these adverse impacts is crucial for improving the performance of such reputation management methods.

4 BGET Approach

BGET is designed as a distributed reputation management approach for P2P networks, in which all peers can effectively distinguish trusted ones from non-trusted ones, choose reliable ones to interact with and resist the manipulation of malicious peers, as shown in Fig. 1.

The cores of BGET consist of three critical points: 1) distinguish trusted peers from non-trusted ones, 2) prevent peers from aggregating around high-reputation peers and 3) provide verifiable reputation management services. The first point is a basic require for reputation management methods and the existing ones all can basically meet this requirement to a certain extent. The ET method is one of the most popular and successful methods. It combines local trust and recommendation trust, which makes it more accurate in the evaluation of reputations of peers. The second one and the third one are the two key facets of BGET approach. As we stated earlier, the mechanism of global ranking and pre-trusted peers will cause the aggregation problem, so we design a group strategy

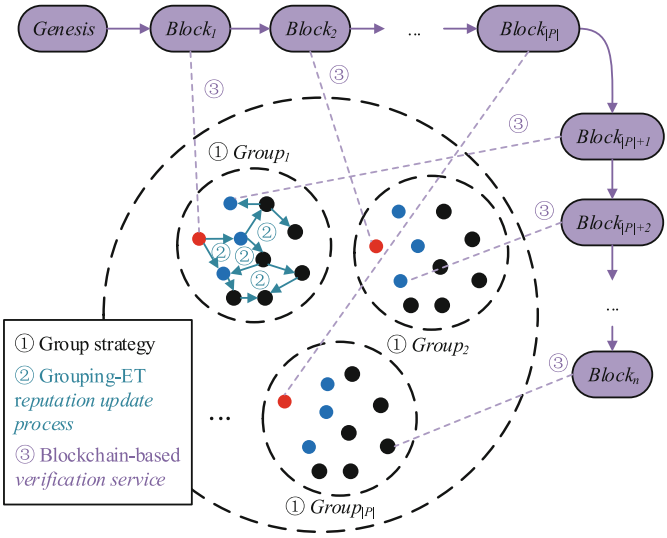

Fig. 1. BGET approach. The red points represent the pre-trusted peers and the blue points represent the peers with high reputation values, while the black ones represent the peers with low reputation values. $|P|$ is the number of pre-trusted peers and n is the number of peers in the network.

to maintain the balance of reputation in the network. The peers in the network will be divided into different group according to the number of pre-trusted peers and the scale of network. By doing this, we can evenly distribute the pre-trusted peers to the network and avoid the aggregation of high-reputation peers. Besides, the slothful peers can only choose to trust the pre-trusted peer in the same group as them. Finally, to decrease the success rate and quality of interact, the non-trusted peers always provide forged data and records to confuse the trusted ones. To go along with this, the blockchain-based reputation verification services are provided in BGET to deal with the problem of fake. With the help of blockchain's tamper resistance and traceability features, BGET can achieve automated periodic verification services, which also adds a supplement safeguard to the non-trusted peer detection process.

The BGET approach mainly includes three processes: 1) using uniform grouping strategy to divide peers to different groups, which can avoid all high-reputation peers gathering together, 2) using grouping-ET reputation updating algorithm to calculate peers' reputation values to ensure dynamic and safe updates of peers' reputations, and 3) providing blockchain-based verification services to detect suspicious behaviors in transactions, such as forged transaction records, etc.

As mentioned above, BGET not only has the advantages of ET, but also achieves uniform distribution of high-reputation peers in the network and

provides reliable verification services to ensure the credibility of peers' reputations updating.

The notations used in our approach are shown in Table 1

Table 1. Notations and descriptions.

Notation	Description
m	Total number of peers in the network
i, j	Peers in the network
P	Pre-trusted group
$\|P\|$	Number of pre-trusted peers
N	Number of peer groups
l_i	Label to mark group i whether include a pre-trusted peer
N_i	The number of peers in group i
A_i	Set of peers which have downloaded files from peer i
B_i	Set of peers from which peer i have downloaded files
α	The probability of pre-trust being selected in ET
s_{ij}	Total number of authentic downloads from peer i to peer j
c_{ij}	Normalized value of s_{ij}
C	Matrix of c_{ij}
n	Number of iterations
t_i	Trust value of peer i
$[t]$	Matrix of t_i
N_{ver}	the number of transactions that need to be verified
Tx_{peer}	represents the total number of peer's transaction records

4.1 Grouping-ET Reputation Updating Algorithm

The reputation calculate algorithm used in BGET approach is an enhanced version of ET, grouping-ET, which combines the uniform grouping strategy and intragroup random walk strategy. In detail, the algorithm process is shown in algorithm 1.

System Initialization. 1) Before the system starts, the parameters, including m, P, N, l_i, A_i and B_i, will be initialized firstly. P represents a pre-trusted peers group, where only pre-trusted peers are present, and $|P|$ is its size. m is the total number of peers in the network, N is the number of groups after the peers is divided, l_i is a label to mark group i whether include a pre-trusted peer, which is 1 if yes and 0 if no. 2) considering the scene of peer j downloads files from peer i, then, there will generate two sets: A_i and B_i. A_i represents the set of peers which have downloaded files from peer i and B_i represents the set of peers from which peer i have downloaded files. The initial values of A_i and B_i are null.

Algorithm 1: Grouping-ET Algorithm.

Input: Number of peers - m; Pre-trusted peers group - P; Number of peer groups - N; Label to mark group i whether include a pre-trusted peer - l_i; Error parameter - ϵ; Set of peers which have downloaded files from peer i - A_i; Set of peers from which peer i have downloaded files - B_i.

Output: Matrix of global reputations of all peers - $[t]$.

1 **for** $x = 1$ *to* N **do**
2 $y = Random(N)$;
3 **if** $peer_x \in P$ **then**
4 **if** $l_y = 0$ **then**
5 $Group_y \leftarrow peer_x$;
6 $l_y = 1$;
7 **else**
8 $y = Random(N)$;
9 **else**
10 $Group_y \leftarrow peer_x$;
11 peer j choose peer i from $Group_j$ to download files;
12 **foreach** $Group_i$ **do**
13 **foreach** *peer* i **do**
14 Query all peers $j \in A_i$ & $j \in Group_j$;
15 **repeat**
16 $t_i^{(k+1)} = \alpha(c_{1i}t_1^{(k)} + c_{2i}t_2^{(k)} + \cdots + c_{mi}t_m^{(k)}) + \alpha p_i$;
17 Send $c_{ij}t_i^{(k+1)}$ to all peers j $\in B_i$;
18 Compute $\delta = \left|t_i^{(k+1)} - t_i^{(k)}\right|$;
19 Wait for all peers $j \in A_i$ & $j \in Group_j$ to return $c_{ji}t_j^{(k+1)}$;
20 **until** $\delta < \epsilon$;
21 return $[t]$.

Uniform Grouping Strategy. The goal of this strategy is to keep the balance of global reputations of different groups and avoid the aggregation of high-reputation peers. The number of groups, N, is a decisive parameter in the process of grouping. It's worth noting that there are three possible values of N: $N < |P|$, $N = |P|$ and $N > |P|$, and we analyze the effects of different values of P through the comparative experiment in Sect. 5.4. According to the result, we set $N = |P|$, which can achieve an optimal balance between the groups. Therefore, the peers in the network will be divided to N groups and each group only has one pre-trusted peers. As shown in the line 1 to 10 of algorithm 1, each pre-trusted peer will be distributed to a different group with the help of label l_i. If group y has included a pre-trusted peer, then $l_y = 1$ and peer x will be distributed to another group with $l_y = 0$. By doing so, the high-reputation peers are no longer clustered together at a single center, but are scattered across different groups.

Intragroup Random Walk Strategy. In original ET, peer j will utilize a random walk way to select a objective peers from the whole network to downloads files. In specific, peer j will select a pre-trusted peer with a probability of α and other peers in the network with a probability of $1 - \alpha$. However, this way will lead to two serious problems: 1) it is a highly uncertain task to choose the value of parameter α, which will cause a significant impact on the probability distribution and 2) if pre-trusted peers accidentally accepted inauthentic files, they will spread them to the whole network faster, since they have more opportunities to be selected. Unlike original ET, our Grouping-ET algorithm utilize intragroup random walk strategy to achieve this goal, which suggest peers to select objective peers from their own group with a random way, as shown in line 11 of algorithm 1 and Fig. 1. In Fig. 1, every circle presents a group and points represent peers in the network. Compare to global random walk strategy of original ET algorithm, our intragroup random walk strategy limits the scope of peers' activities, which can effectively limit malicious peers collaborate with other malicious peers in other groups. In this way, we not only eliminate the influence of parameter α, but also stop the spread of inauthentic files and improve the effectiveness of reputation updating. Next, peer j selected a peer i in this way and downloaded some files from peer i. Then, peer j will be added into set A_i. In the same way, peer i will also maintain a set B_i that it have downloaded files from.

Reputation Updating. As shown algorithm 1, except for no longer using the original parameter α in ET, the other processes of reputation updating is roughly the same as ET. As demonstrated in intragroup random walk strategy, to eliminate the uncertainty of parameter α, we redefine α to a random parameter and let $\alpha = \frac{1}{N_i}$, where N_i represents the number of peers in group i. Then, the updating process is shown in line 12 to 20 of algorithm 1. For every transaction, the reputation updating process will be invoked to update the reputation of both parties involved in the transaction. It is worthy noting that, since the improvement of uniform grouping strategy and intragroup random walk strategy, our method has lower complexity than original ET algorithm, which is discussed in Sect. 5.1.

Blockchain-Based Verification Services. As stated in uniform grouping strategy and intragroup random walk strategy, transactions are limited in peers' own groups. There is no doubt about that honest peers will adhere strictly to these regulations, but malicious peers may take risks. These regulations can only limit the behaviors of honest ones, the malicious ones have no reason to comply. Therefore, it is necessary to develop rules that can fairly limit the behaviors of all members. In view of this, the blockchain-based verification services will be an effective way to ensure the correctness and effectiveness of transactions. By consulting the non-manipulable historical transaction records stored in blocks, BGET can discover different violations such as falsification of data or transaction records and so on.

4.2 Blockchain-Based Verification Services

Another key point of BGET is to provide verification services, which can prevent dishonest peers from engaging in fraudulent behaviors. To achieve this, BGET designs several verification services based on blockchain for reputation updating, the process is shown in Fig. 2.

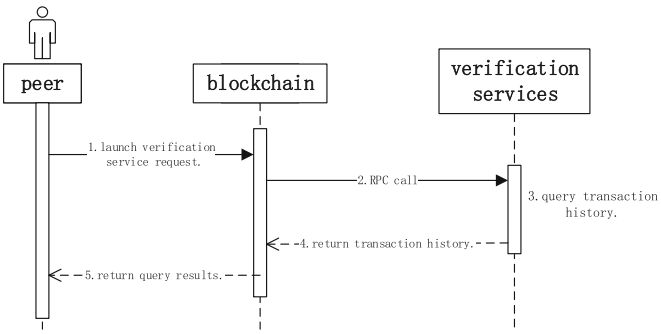

Fig. 2. Verification services based on blockchain.

For P2P environment, there are a large number of peers and frequent transactions. To verify every transaction is not a reasonable way as it will result in a significant waste of time and computing resources. Therefore, in BGET, the verification service is set to activate once every 100 times transactions. The detail verification polices are as follows:

- For peers who have no abnormal transactions, in order to ensure the efficiency of the system, high-reputation peers of each group can be exempted from verification. For those who have abnormal transactions or low-reputation values, they need to follow the latter policies.
- For new members, they only have a handful of historical transactions and just need to verify a small number of transactions at random.
- For slothful peers, who are honest but rarely engage in transactions, also need to verify a small number of transactions at random.
- For most of normal peers, the number of transactions that need to be verified is: $N_{ver} = (1 - t_{peer})Tx_{peer}$, t_{peer} represents the reputation value of the peer and Tx_{peer} represents the total number of peer's transaction records. In order not to affect efficiency, there is no need to verify all records every time, but only to verify the most recent records, since the earlier records have already been verified. Here, $Tx_{peer} = 100$ which means that only N_{val} out of the last 100 records need to be verified each time.
- Penalties. If a peer is found to provided inauthentic resources to other peers, it could be a malicious peer or just a victim who accept inauthentic resources from others. In this situation, the peers will be deducted twice the points

that should be earned for this transaction to reduce the likelihood of it being selected next time. If a peer is found to have forged its transaction records to improve its rating in the past, it must be a malicious node. Then, the reputation of this peer will be clear to zero.

According to the polices of BGET, peers must choose the peers in the same group with them to interact with. If the verification services find a peer has transaction records that completed with peers in other groups, the peer will be recognized as malicious peer and these downloads will also be marked as inauthentic downloads. Besides, blockchain has the characteristic of traceability. When inauthentic files are found, the source can be traced and the inauthentic files can be cleaned up as much as possible. With the help of these verification services, all members in the network will have a more fair environment, where peers can trade with each other securely and the updating of reputation also be more credibly.

5 Performance Evaluation

In this section, we analyse the time complexity of BGET and simulate several scenes to study its behaviors. Besides, we compare the performance of BGET to a simple P2P network where no reputation management system is implemented, original ET [7] and HonestPeer [12], a popular variant of ET.

5.1 Algorithm Complexity and Convergence Overhead

According to algorithm 1, the complexity of BGET is mainly dependent on the calculation of ${t_i}^{(k+1)}$ in formula (2) and the convergence overhead. Unlike the original ET algorithm which needs to be iterated over the whole network, BGET is sampler. In original ET algorithm, although $C^{\top}$ in formula (1) is sparse, since most of c_{ij} have a value of 0, it still needs to traverse all peers in the network every loop. However, BGET approach only needs to converge within the group where the peers belong to, greatly reducing the convergence time. Moreover, the convergence overhead is also affected by impact of number of loop. Same as above, the number of iterations required within each group is also smaller than ET. Hence, the algorithm complexity and convergence overhead of BGET are superior to the original ET algorithm.

5.2 Simulation Setup

Our evaluation is based on the simulation execution with many simulation cycles. In every simulation cycle, we build a simulation network with different number of peers, include honest peers and malicious peers, (as demonstrated in Table 2). Honest peers always provide authentic downloads, while malicious peers always provide inauthentic downloads. Every peer has two status: query and standby. In query status, peer i will issue a request and wait for responses. Other peers

in standby status receive the request and return acknowledgments to indicate that they can provide the service. The query peers will choose a service provider j based on BGET approach from the set of hits. Then peer i gives a score of 0 to 1 based on the quality of the downloaded files. At the end of each simulation cycle, the global reputation values of all peers in the network will be calculated according to our BGET algorithm. These results will be utilized in the upcoming simulation cycles for the next selection of service provider.

All simulations are carried on an Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-71-generic x86_64) laptop with one 16 core CPU. The software tools we used is PyCharm 2020.1, we simulate all the peers and blockchain in it to evaluate the performance of BGET.

It is noting that although the experimental scenario is set to file-downloading, the BGET approach can be adapted to various transaction scenario, such as e-commerce, file sharing, social networks, and so on.

Table 2. Configuration of the simulation parameters.

Parameter	Value
m	500-3500 (step=500)
$\|P\|$	5/10/15
N	5/10/15
% Proportion of malicious peers	10%-100% (step=10%)
# Simulation cycles	25
# Runs for each experiment	5

5.3 Evaluation Criteria

In file-sharing scene, a commonly used evaluation criteria is the rate of inauthentic downloads, which reflects the whole health of the network in decreasing the frequency of downloads of fraudulent files. The smaller the rate of inauthentic downloads, the better the system performance. The rate of inauthentic downloads is defined as:

$$\text{Rate of inauthentic downloads} = \frac{\#\text{downloads of fraudulent files}}{\#\text{downloads done by all peers}} \tag{3}$$

5.4 Results and Analysis

In our BGET approach, the number of the pre-trusted peers, $|P|$, is an important parameter of uniform grouping strategy. There are three situations: (i) $N < |P|$, the number of pre-trusted peer divided into each peer group will be unbalanced and greater than 1. (ii) $N = |P|$, the distribution of pre-trusted peers is balance and all groups have the same number of pre-trusted peers, all of which are 1. (iii)

$N > |P|$, some groups will not contain pre-trusted peers, which will cause them to lose an advantage in competition. To evaluate the impact of these settings on the system performance, we fix the value of N to 10, number of peer to 1000, and conduct comparative experiments based on different values of $|P|$ and proportion of malicious peers. The result is shown in Fig. 3.

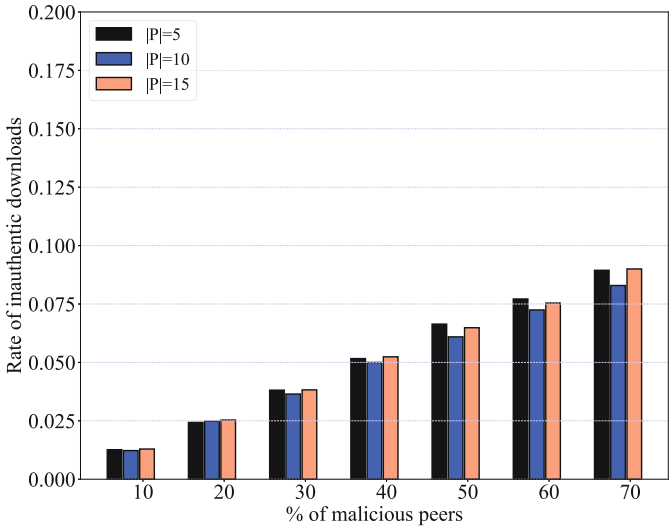

Fig. 3. The effect of $|P|$ on BGET approach. $per\_num = 1000$.

In Fig. 3, when the proportion of malicious peers is less than 30%, the rates of inauthentic downloads in the system of these three conditions are almost the same. As the malicious peers proportion increases, the gap of the inauthentic downloads rates also chases up. To be specific, when $N = |P|$, the inauthentic rate is minimum, and it is slightly better when $N < |P|$ than when $N > |P|$. Hence, setting $N = |P|$ is more conducive to the security assurance of the system, especially when the proportion of malicious peers is larger. In the upcoming experiments, the values of N and $|P|$ both are set to 10 to maintain the balance of uniform grouping.

Then, to evaluate the performance of BGET approach under different threat environments, we consecutively conduct different simulation experiments based on the different configuration parameters, as shown in Table 1. The results are shown in Fig. 4.

Figure 4 illustrates the variation trends of inauthentic downloads rates for different methods under different situations that with different number of peers. Only the simple method experiences a spike when the total number is increased from 500 to 1000, and continue to maintain a growth state. Other than that, other methods all remain a stable state, that is, these methods are insensitive

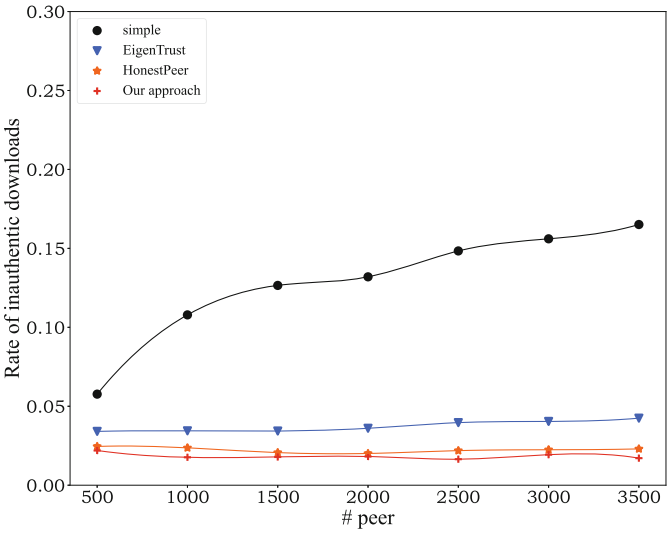

Fig. 4. The effect of peers number to different methods.

to the expansibility of network. So, BGET fairly meets the needs of large scale distributed system.

Finally, to evaluate the impact of different malicious rates on the effectiveness of our BGET approach, we simulate different scenarios with different malicious rates. The result is shown in Fig. 5.

Figure 5 shows the reactions of different methods when the proportion of malicious peers increases. The ET and HonestPeer methods have similar growth curve, and the HonestPeer is superior to ET. When the proportion of malicious peers exceeds 65%, both of them have an inauthentic download rates over 60%, which is higher than the simple method. It indicates that the performance of these methods will be awful when most peers in the network are malicious peers. As you can imagine, the user experience in this case will be terrible. By comparison, the inauthentic downloads rate of our BGET approach grows steadily with a fixed and small growth rate and only shows slight reaction to the increase of malicious peers when the rate below 80%, and the inauthentic download rate can be stabilized at around 10%. However, after the malicious rate exceeds 80%, the rate of inauthentic downloads of BGET also experience a dramatic increase. That is to say, the malicious rate of 80% is the limit value at which our method can maintain acceptable performance, which is far better than ET and HonestPeer. Of course, this is a reasonable phenomenon. When almost all members in a network are malicious, even the best method cannot achieve ideal performance. It's worth noting that even if there are no malicious peers in the network, inauthentic downloads still account for almost 5% - this accounts for mistakes users make when creating and sharing a file, e.g., by providing the wrong meta-data or creating and sharing an unreadable file [7].

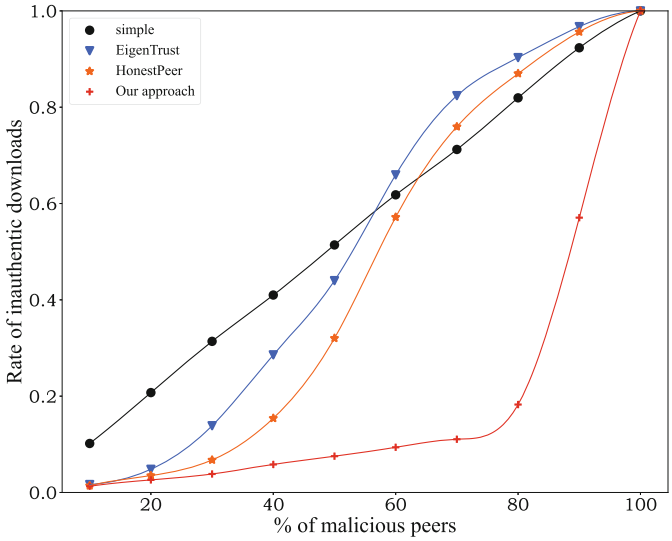

Fig. 5. The effect of proportion of malicious peers to different methods. *peer_num* = 1000.

On the whole, our BGET approach has good scalability when the scale of network increases and responds well to different size of peers and malicious rates. BGET can maintain a low inauthentic downloads rate when the malicious rate is lower than 80%. Compare to methods such as ET and HonestPeer, BGET exhibit strong robustness when facing harsh environments with a large number of malicious peers.

6 Conclusion

In this paper, we expounded why peers will gather around the pre-trusted peers and high-reputation peers, and this aggregation will drive the network towards centralization, which is contrary to the original intention of P2P networks. Besides, the pre-trust mechanism will lead to uncontrolled spread of inauthentic downloads. To address these problems, we proposed a BGET reputation management approach for P2P networks. In BGET, we utilized uniform grouping strategy to prevent the aggregation of high-reputation peers, put forward a grouping-GT reputation updating algorithm and design some reliable verification services based on blockchain to avoid the fraud of malicious peers. Finally, we employed several simulations to evaluate the effectiveness of BGET. The results showed that BGET not only exhibits strong scalability for the increasing number of peers, but also strong robustness for the increasing proportion of malicious peers. For the success rate of downloading authentic files, BGET can keep the rate of inauthentic downloading at around 10% when the malicious rate below 80%, which is far better than ET and HonestPeer.

In future work, we will focus on evaluation parameters and different malicious attacks to improve the robustness of our approach, and study the schemes that we could integrate our approach into the consensus process of some mature blockchain networks, such as Hyperledger, EOS and so on.

References

1. Aberer, K., Despotovic, Z.: Managing trust in a peer-2-peer information system. In: Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA, November 5-10, 2001, pp. 310–317. ACM (2001). https://doi.org/10.1145/502585.502638
2. Alhussain, A., Kurdi, H.A.: EERP: an enhanced Eigentrust algorithm for reputation management in peer-to-peer networks. Procedia Comput. Sci. **141**, 490–495 (2018). https://doi.org/10.1016/j.procs.2018.10.137
3. Can, A.B., Bhargava, B.K.: SORT: a self-organizing trust model for peer-to-peer systems. IEEE Trans. Dependable Secur. Comput. **10**(1), 14–27 (2013). https://doi.org/10.1109/TDSC.2012.74
4. Chiluka, N., Andrade, N., Gkorou, D., Pouwelse, J.A.: Personalizing Eigentrust in the face of communities and centrality attack. In: IEEE 26th International Conference on Advanced Information Networking and Applications, AINA, 2012 , Fukuoka, Japan, March 26-29, 2012, pp. 503–510. IEEE Computer Society (2012). https://doi.org/10.1109/AINA.2012.48
5. Govindaraj, R., Priya, G., Chowdhury, S., Kim, D., Tran, D., Le, A.: A review on various applications of reputation based trust management. Int. J. Interact. Mob. Technol. **15**(10), 87 (2021). https://doi.org/10.3991/ijim.v15i10.21645
6. Jøsang, A., Golbeck, J.: Challenges for robust trust and reputation systems. In: 5th International Workshop on Security and Trust Management (STM 2009) (2009)
7. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The Eigentrust algorithm for reputation management in P2P networks. In: Proceedings of the Twelfth International World Wide Web Conference, WWW 2003, Budapest, Hungary, May 20-24, 2003, pp. 640–651. ACM (2003). https://doi.org/10.1145/775152.775242
8. Karjalainen, R.: Governance in decentralized networks (2020). https://doi.org/10.2139/ssrn.3551099
9. Khan, Z.A., Abbasi, U.: SORT: A self-organizing trust model for peer-to-peer systems. Electronics **9**(3), 415 (2020). https://doi.org/10.3390/electronics9030415
10. Kouicem, D.E., Imine, Y., Bouabdallah, A., Lakhlef, H.: Decentralized blockchain-based trust management protocol for the internet of things. IEEE Trans. Dependable Secur. Comput. **19**(2), 1292–1306 (2022). https://doi.org/10.1109/TDSC.2020.3003232
11. Kremenova, I., Gajdos, M.: Decentralized networks: the future internet. Mob. Netw. Appl. **24**, 2016–2023 (2019). https://doi.org/10.1007/s11036-018-01211-5
12. Kurdi, H.A.: HonestPeer: an enhanced Eigentrust algorithm for reputation management in P2P systems. J. King Saud Univ. Comput. Inf. Sci. **27**(3), 315–322 (2015). https://doi.org/10.1016/j.jksuci.2014.10.002
13. Lee, J., Becker, K.: Organizational usage of social media for corporate reputation management. J. Asian Financ. Econ. Bus. **6**(1), 231–240 (2020). https://doi.org/10.13106/jafeb.2019.vol6.no1.231

14. Li, M., Guan, Q., Jin, X., Guo, C., Tan, X., Gao, Y.: Personalized pre-trust reputation management in social P2P network. In: 2016 International Conference on Computing, Networking and Communications, ICNC 2016, Kauai, HI, USA, February 15-18, 2016. pp. 1–5. IEEE Computer Society (2016). https://doi.org/10.1109/ICCNC.2016.7440695
15. Lin, Y.J., Yang, H.W., Yang, C.C., Lin, W.: A traceable and fair transaction mechanism for digital rights management on P2P networks. J. Internet Technol. **14**(7), 1043–1051 (2013). https://doi.org/10.6138/JIT.2013.14.7.04
16. Liu, Y., Wang, J., Yan, Z., Wan, Z., Jäntti, R.: A survey on blockchain-based trust management for internet of things. IEEE Internet Things J. **10**(7), 5898–5922 (2023). https://doi.org/10.1109/JIOT.2023.3237893
17. Lu, K., Wang, J., Li, M.: An Eigentrust dynamic evolutionary model in P2P file-sharing systems. Peer-to-Peer Netw. Appl. **9**, 599–612 (2016). https://doi.org/10.1007/s12083-015-0416-1
18. Resnick, P., Kuwabara, K., Zeckhauser, R., Friedman, E.: Reputation systems. Commun. ACM **43**(12), 45–48 (2000). https://doi.org/10.1145/355112.355122
19. Su, S., Tian, Z., Liang, S., Li, S., Du, S., Guizani, N.: A reputation management scheme for efficient malicious vehicle identification over 5G networks. IEEE Wirel. Commun. **27**(3), 46–52 (2020). https://doi.org/10.1109/MWC.001.1900456
20. Tian, Z., Gao, X., Su, S., Qiu, J., Du, X., Guizani, M.: Evaluating reputation management schemes of internet of vehicles based on evolutionary game theory. IEEE Trans. Veh. Technol. **68**(6), 5971–5980 (2019). https://doi.org/10.1109/TVT.2019.2910217
21. Walsh, K., Sirer, E.G.: Fighting peer-to-peer SPAM and decoys with object reputation. In: Proceedings of the 2005 ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems, P2PECON 2005, Philadelphia, Pennsylvania, USA, August 22, 2005, pp. 138–143. ACM (2005). https://doi.org/10.1145/1080192.1080204
22. Wu, X., Liang, J.: A blockchain-based trust management method for internet of things. Pervasive Mob. Comput. **72**, 101330 (2021). https://doi.org/10.1016/j.pmcj.2021.101330
23. Xiao, Y., Liu, Y.: BayesTrust and Vehiclerank: constructing an implicit web of trust in VANET. IEEE Trans. Veh. Technol. **68**(3), 2850–2864 (2019). https://doi.org/10.1109/TVT.2019.2894056
24. Xiong, L., Liu, L.: Peertrust: supporting reputation-based trust for peer-to-peer electronic communities. IEEE Trans. Knowl. Data Eng. **16**(7), 843–857 (2004). https://doi.org/10.1109/TKDE.2004.1318566
25. Xue, W., Liu, Y., Li, K., Chi, Z., Min, G., Qu, W.: DHTrust: a robust and distributed reputation system for trusted peer-to-peer networks. Concurr. Comput. Pract. Exp. **24**(10), 1037–1051 (2012). https://doi.org/10.1002/cpe.1749
26. Yang, Z., Yang, K., Lei, L., Zheng, K., Leung, V.C.M.: Blockchain-based decentralized trust management in vehicular networks. IEEE Internet Things J. **6**(2), 1495–1505 (2019). https://doi.org/10.1109/JIOT.2018.2836144
27. Liu, Y., Wu, L., Li, J.: Peer-to-peer (p2p) electricity trading in distribution systems of the future. Elect. J. **32**(4), 2–6 (2019). https://doi.org/10.1016/j.tej.2019.03.002
28. Yu, B., Singh, M.P.: A social mechanism of reputation management in electronic communities. In: Klusch, M., Kerschberg, L. (eds.) CIA 2000. LNCS (LNAI), vol. 1860, pp. 154–165. Springer, Heidelberg (2000). https://doi.org/10.1007/978-3-540-45012-2_15
29. Zhou, R., Hwang, K.: PowerTrust: a robust and scalable reputation system for trusted peer-to-peer computing. IEEE Trans. Parallel Distrib. Syst. **18**(4), 460–473 (2007). https://doi.org/10.1109/TPDS.2007.1021