The ISC Int'l Journal of Information Security

January 2015, Volume 7, Number 1 (pp. 59–74)

http://www.isecure-journal.org

GGRA: A Grouped Gossip-based Reputation Aggregation Algorithm

Safieh Ghasemi Falavarjani [1,*], Behrouz Tork Ladani [1], and Simin Ghasemi [2]

[1] *Department of Computer Engineering, University of Isfahan, Isfahan, Iran*

[2] *Department of Computer Engineering, Payame Noor University (PNU), Iran*

ARTICLE INFO.

Article history:
Received: 21 April 2014
Revised: 31 May 2015
Accepted: 15 July 2015
Published Online: 20 July 2015

Keywords:
Gossip Algorithm, Peer to Peer Network, Reputation Aggregation, Group Based Reputation, Group Based P2P Systems.

ABSTRACT

An important issue in P2P networks is the existence of malicious nodes that decreases the performance of such networks. Reputation system in which nodes are ranked based on their behaviour, is one of the proposed solutions to detect and isolate malicious (low ranked) nodes. *GossipTrust* is an interesting previously proposed algorithm for reputation aggregation in P2P networks based on the concept of gossip. Despite its important contribution, this algorithm has deficiencies especially with high number of nodes that leads to high execution time and low accuracy in the results. In this paper, a grouped Gossip based Reputation Aggregation (*GGRA*) algorithm is proposed. In *GGRA*, *GossipTrust* is executed in each group between group members and between groups instead of executing in the whole network. Due to the reduction in the number of nodes and using strongly connected graph instead of a weakly one, gossip algorithm in *GGRA* is executed quickly. With grouping, not only reputation aggregation is expected to be more scalable, but also because of the decrement in the number of errors of the gossiped communication, the results get more accurate. The evaluation of the proposed algorithm and its comparison with *GossipTrust* confirms the expected results.

1 Introduction

P2P Network is a network that all nodes are both client and server. One of the aims of these networks is to increase the information accessibility. P2P networks have been used in many applications including file sharing and military communication systems. One of the most important challenges in such type of networks is security implications which arise from abusing the trust between peers [1]. Dealing with malicious nodes is the substantial point of this challenge. In this problem, each peer can usually attend or leave the network without any central control and malicious behaviours like sharing virus files can affect the whole network easily and widely. The existence of this type of issues caused the creation of trust and reputation systems. These systems maintain the information about the previous behaviour of the peers and decide on the next contacts based on this information. Reputation and Trust systems use trust and reputation concepts for facing and managing malicious behaviours. Although there are many definitions for *trust* concept, following is a well-defined explanation of it: "Trust reflects the belief or confidence or expectations on the honesty, integrity, ability, availability

* Corresponding author.
Email addresses: saghasemif@eng.ui.ac.ir (S. Ghasemi Falavarjani), ladani@eng.ui.ac.ir (B. Tork Ladani), s.ghasemi@zanjanpnu.ac.ir (S. Ghasemi).

ISSN: 2008-2045

and quality of service of target node's future activity or behaviour" [2]. In other words, *trust* refers to the opinion of each peer about the behaviour of the others according to its previous experiences. This opinion usually is declared with a numerical value. Reputation systems help peers to choose reliable sources for transactions by aggregating different trust values and ranking peers based on their global reputations. Reputation aggregation methods as a part of reputation systems have been developed for gathering information about previous transactions and behaviours of the peers.

The main problems in P2P systems are how to gather, spread and control information which is needed for computing reputation at each node. In these networks, a desired reputation system should have minimum overhead in infrastructure, computation, storage and number of messages [1, 3]. Fault tolerance, scalability and accuracy in determining malicious peers are the other features which an acceptable reputation system should have.

Gossip algorithm is a suitable method for quick and correct reputation aggregation in P2P systems [3–5]. Gossip based data aggregation is inspired by gossip dissemination in social relations. In this method, in each step or time period, each node communicates and exchanges information with one or more nodes that are randomly chosen which leads to approximate distribution of information to all nodes [6, 7]. Gossip algorithm is simple and has remarkable features including robustness, fault tolerance and scalability [7]. However, it imposes pretty much overhead in the networks with large number of peers. *GossipTrust* [3] is a noteworthy method which uses Gossip algorithm. In this method, peers share their local trust values with their random neighbors so that the values converge to global reputation for every peer. Despite *GossipTrust* contribution in increasing the scalability, by increasing in the number of peers, not only computation time of the reputation aggregation increases, but also accuracy and convergence of the reputation values confront some obstacles [8]. *GossipTrust* uses the *Kempe et. al*'s algorithm [7] for gossiping. The *Kempe* algorithm has been proven to be correct when network is strongly connected [6]. This is while P2P networks are not strongly connected and it is known that structure of P2P networks shows power-law properties [9, 10]. This inconsistency reduces the accuracy of *GossipTrust* and increases the convergence time of the reputation data.

In P2P networks, dynamic nature and large number of nodes lead to some issues for completely distributed algorithms. Management, aggregating and computation of the reputation information are some of these issues. *GossipTrust* has been designed for completely distributed networks. But, in this method, by growing the number of nodes, high overhead causes reduction in the number of transactions and participation of the nodes. One possible solution to get out from completely distributed systems is to consider different levels for nodes based on their capabilities and to use a semi-central reputation system. In semi-central systems, disadvantages of both distributed and central systems decrease and advantages of both systems can be used. Semi-central management is one of the existing methods for managing P2P networks as the proposed second generation architecture in [11]. This architecture shows that selection of semi-central system according to different capabilities of nodes will improve performance.

One way for using advantages of semi-central architecture is to divide peers in groups according to different criteria. In P2P networks, there are some methods for establishing groups based on interests and friendship between peers. The aim of grouping is to extend the security, increase participation of members and accomplish faster search for the required resources. When groups are constructed based on peer's interests, requests for services or resources are handled inside groups. Therefore, lots of useless searches which only waste the network resources will be prevented [12, 13]. Also, there is no need to compute reputation of all peers in the system. Each peer just tracks reputation of the nodes inside the group. Meanwhile, group formation is an important point in terms of security as discussed in some works such as [14–16]. Group manager can set some security rules to protect members against malicious nodes. In this way, members participate more in defending from their group against other groups and probable attacks. Furthermore, the existence of society and feeling of belonging to the group can prevent lots of malicious activities. Evaluation of trust in grouping the peers has been already studied as *GTrust* system by *Ezhei et al.* in [17].

The problem that is considered in our paper is the underlying computation and communication overhead in gossip based reputation systems in P2P networks. According to the previous explanation, to propose a faster, more secure and more accurate reputation aggregation model in P2P networks, grouping and semi-centralization concepts are used in this paper. The aim is to present a reputation system using groups and based on gossip to make distribution and aggregation more scalable. Specifically with choosing *GossipTrust* as reputation aggregation algorithm in P2P network, endeavors were made to increase the performance of reputation aggregation.

In the proposed algorithm named Grouped Gossip

based Reputation Aggregation (*GGRA*), *GossipTrust* is executed in every group separately and simultaneously. Therefore, the number of participating nodes in each *GossipTrust* execution decreases. On the other hand, *GossipTrust* performance in groups with high clustering coefficient is more efficient. Due to the reduction in the number of nodes and also changing weakly connected network to strongly connected one before aggregation, the speed of reputation aggregation increases. Therefore, number and storage space of messages and also computation time for aggregation decrease. With grouping, not only reputation aggregation is more scalable, but also error of the gossiped results decreases. In this model, representatives of groups are used as middle agents between distributed nodes (members of group) and other groups' representatives. Actually representatives present a semi-central role in the system. While in completely distributed P2P networks all nodes are considered equal and the advantage of powerful nodes is ignored, in a semi-central approach, nodes are used based on their capabilities. The above are the potential advantages that motivate the authors to propose the *GGRA*.

The remainder of the paper is organized as follows. In Section 2, literature of the proposed algorithm and most important related works are specified. Also main concepts of the paper including P2P network, reputation and group are discussed in this section. In Section 3, model assumptions, executive phases, time and message complexities of the proposed algorithm are explained. The simulation of the proposed algorithm followed by different evaluations and comparisons with *GossipTrust* are described in Section 4. Finally, Section 5 concludes the paper.

2 Related Work

Reputation systems provide a method for predicting the quality and reliability of transactions in the future. The main problem in reputation management is how to deal with malicious behaviors. The most general way is to use the peers' recommendation about the providers. Peers rate those who have previously communicated with them. Peers, who are requesting a service, establish trust through analyzing different trust values which others have been given to the providers [1]. P2P reputation management systems should be distributed, efficient, scalable, reliable, and secure in computation, storage and spreading the trust values. There are some useful surveys about the category of reputation, trust systems and security threats followed by their comparison [1, 2, 18, 19]. The work in [19] is a survey which introduces a remarkable taxonomy for reputation systems, along with a reference model for reputation systems as well as the comparison of existing reputation researches and the deployed reputation systems. As one of the recent works on reputation systems, Secure and Effective distributed P2P Reputation System proposed by *Srikanth* and *Madhuri* in [20] can be mentioned. In this model, Self-certification, an identity management mechanism, and a cryptographic protocol to exchange trust between the peers are used. This method facilitates generation of secure reputation data in a P2P network, in order to expedite detection of rogues.

The core of the proposed algorithm in this paper is the gossip algorithm. The most important and related gossip based algorithms are following. *Kempe et al.* [7] proposed some gossip based algorithms for data aggregation. *Jelasity et al.* [6] also proposed the push-pull gossip algorithm. In not strongly connected networks, the *Jelasity* algorithm works better than the *Kempe* algorithm for data aggregation. Gossip algorithms are also used by *Bachrachet et al.* [4] for reputation aggregation. *DifferentialGossipTrust* [5] is a gossip based algorithm for aggregating reputation specifically in power-law networks. Instead of simple push gossip algorithm such as *Kempe*, it suggests a differential push gossip algorithm. Nodes uses different number of pushing based on the degrees of themselves and their neighbors. The authors in [3] proposed the *GossipTrust* algorithm which is used as the basic method for reputation aggregation in the proposed model in this paper. We will review this algorithm in detail in the next section.

To increase efficiency of communications, social communication models and virtual social networks are increased recently. P2P networks are not exception to this rule. For solving some problems of P2P systems, social based or Group based solutions are suggested. The proposed *GGRA* algorithm also uses grouping concept to reach its mentioned goals. Previously, some group based models are suggested. One of the proposed models is grouping based on peers' interests [12, 13, 17, 21]. Using this kind of grouping, requests for resources (services) are most related to their own groups. Therefore, searching related groups for finding owners of target resources is usually enough. This way, searching for resources can be done more quickly, so most of the useless searching time and traffic can be saved. Other researches based on groups are done with the aim of setting secure policies and increasing security in networks [16, 22, 23]. Since grouped P2P networks are more practical and peers usually trust to their groups, reputation systems have been created based on groups. Further, existence of different P2P groups requires reputation system, which is able to obtain reputation of peers based on features of groups. Authors in [16] have proposed a trust and reputation model based on peer group as *GroupTrust*. In this model, reputation is computed based on the

similarities of services' contexts and the similarity of groups. Also, the trust values of those who have communicated with a peer and are in the same group are considered. *GroupRep* [22] is another group based reputation system for P2P networks which proposed a filtering cluster algorithm to filter unfair rating provided by malicious nodes. *GTrust* [17] provides better trust using group membership and trust propagation between unknown persons. The other model proposes a solution for access control [14]. In this model, selection of service provider and also response to requests are based on reputation of peers' groups. *GARM* [15] model divides peers which have the same resources in the same groups. In *GARM*, each pre-trusted node in each group chooses the provider. Pre-trusted node also makes an anonymous path from provider to requester for sending the service. In this model peers compete for increasing their own reputation. In *EigenGroupTrust* [23], each group has a leader who is a pre-trusted node. Each peer uses credential and trust delegation to ensure reliability of resources. In this algorithm at first *EigenTrust* [24] algorithm is executed in each group and then *EigenTrust* is performed between leaders. Note that *EigengroupTrust* has not considered gossip as a method for aggregating in distributed P2P networks. *ILGT* [25] and *GRAT* [8] are partly similar models which both use *GossipTrust* and try to use group formation to improve *GossipTrust* scalability.

ILGT and *GRAT* are the most related works to *GGRA* algorithm, but the concept of group in these two works are different from *GGRA* and they have some deficiencies. In *ILGT*, peers can easily manipulate their own list and collusive groups can easily be formed. Also, while for getting accurate reputation based on *GossipTrust,* reputation of all peers are needed to be calculated simultaneously, in *ILGT* only reputation of the provider peers are computed in each group. In general, the assumptions of *ILGT* and *GRAT* affect accurate execution of *GossipTrust* and these assumptions are not justifiable.

2.1 GossipTrust Algorithm

In this Section *GossipTrust* algorithm [3] is described in detail. Table 1 describes the used variables in this algorithm. The algorithm works as follows:

In a P2P network with n nodes, after each transaction between two nodes, the transaction receiver expresses its trust to the sender by a number. For aggregating reputation, each node i normalizes all its own trust values by using Equation 1:

$$S_{ij} = t_{ij} / \sum_{k=1}^{n} t_{ik} \tag{1}$$

In this equation, the trust value is mapped to a relative portion of summation of trust values from i to other nodes. In *GossipTrust,* there are some aggregation cycles and each aggregation cycle consists of some gossip steps (Figure 1). In this algorithm, each node i has a pair as (x_{ij}, w_{ij}) for each node j in the network which x_{ij} is the gossip reputation and w_{ij} is the gossip weight. In the first step of the algorithm, x_{ij} is equal to s_{ij} as the normalized trust value, w_{ij} is zero and w_{ii} is one for all nodes in the network. In the next steps, x_{ij} and w_{ij} are updated by Equation 2 and Equation 3:

$$x_{ij}(k_i + 1) = \sum_{i'=1}^{r} x_{i'j}(k_i) \tag{2}$$

$$w_{ij}(k_i + 1) = \sum_{i'=1}^{r} w_{i'j}(k_i)$$

$$RGossiped_{ij}(k_i) = \frac{x_{ij}(k_i)}{w_{ij}(k_i)} \tag{3}$$

In each step, each node i sends half of pairs values i.e. $(1/2\ x_{ij}, 1/2\ w_{ij})$ to a randomly selected neighbor and also to itself. Each node aggregates all the received pairs in the previous step with its own pairs by Equation 2. Aggregated reputation in each step k_i for each j is obtained by Equation 3. This value is the final reputation that node i has calculated by receiving and updating different gossip values about j in different steps till step k_i.

Gossip steps as shown in Figure 1 are repeated until all the nodes converge to the reputation value of nodes. Convergence testing is done through Equation 4:

$$\forall_{i,j} \mid RGossiped_{ij}(k_i) - RGossiped_{ij}(k_i - 1) \mid < \varepsilon \tag{4}$$

RGossiped in the last step in cycle k', is considered as the final reputation of the nodes in that cycle and is shown by $R(k')$. The aggregated reputation of each node j at the end of each cycle is equal to the weighted sum of the trust values about j according to the Equation 5:

$$R_i(k'_i) = S^T \times R_i(k'_i - 1) \tag{5}$$

After convergence of steps in a cycle, next cycle starts and this process continues until all reputation values converge with previous cycle which are tested by Equation 6:

$$\forall_{i,j} \mid R_{ij}(k'_i) - R_{ij}(k'_i - 1) \mid < \delta \tag{6}$$

After the convergence of cycles, the aggregated reputation matrix R_i which is obtained by i is the converged global reputation vector of matrix S. The number of steps in a cycle (g), and also the number of cycles to reach final convergence (d), are upper bounded by $O(log(n))$. Figure 2 shows an example of gossip aggregation in *GossipTrust* algorithm. Part

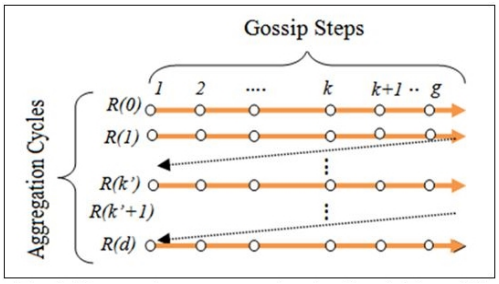

Figure 1. Reputation aggregation in *GossipTrust* [3]

Table 1. Description of the used variables

Variables	Description
t_{ij}	Trust value of i to j
m	Number of members in the group
S_{ij}	Normalized trust of i to j
x_{ij}	Gossip value
w_{ij}	Weight of gossiping
k_i	Node i gossiping step
k'_i	Node i aggregation cycle
$R(k'_i)$	Converged reputation matrix of cycle k'_i
$R_i(k'_i)$	ith column of the matrix $R(k'_i)$
δ	Error threshold for testing convergence of the cycles
$RGossiped_{ij}(k_i)$	Reputation by i in k_i for each j
ε	Error threshold for testing convergence of the gossip steps
d	Number of the cycles that is executed for convergence
p	Set of power nodes
P_{ij}	Extra reputation of power node j considered by i
α	Coefficient for increasing effectiveness of power nodes
$R^P(k'_i)$	Reputation matrix obtained by i in k'_i using power nodes

(a) is the transactions of three nodes i, p and q at the first step. Part (b) indicates gossip pairs of all nodes in the next step.

In P2P networks, new nodes usually connect to the most reputable and pre-trusted peers (power nodes). Using power nodes in the reputation calculation increases the speed of convergence and reduces the collusive effect [3, 24]. Equation 7 shows consideration of power nodes in calculating the reputation [26]:

$$P_{ij} = \begin{cases} \frac{1}{|p|} & if\ j \in p \\ 0 & otherwise \end{cases} \tag{7}$$

$$R_i^p(k'_i) = (1-\alpha)S^T \times R_i(k'_i - 1) + \alpha p$$

In this way, each power node has initial equal $1/|p|$ reputation and other nodes have no reputation. After calculating reputation, reputation of power nodes are added by predefined α factor and reputation of other nodes are reduced by 1-α.

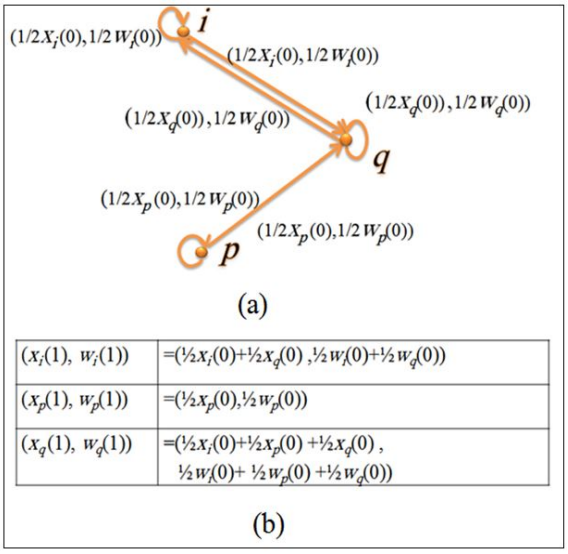

Figure 2. Gossip aggregation in a network with three nodes (a) Transactions in the first phase (b) Gossip pairs in the next phase

3 The Proposed Algorithm – GGRA

Our proposal is built upon the *GossipTrust* algorithm and the grouping concept. In *GGRA*, reputation aggregation is done through executing of *GossipTrust* in and between groups. This section describes the proposed algorithm in detail.

3.1 Basic Assumptions

In the proposed algorithm, it is assumed that there exist some distinct established groups in the network. Each peer is member of only one group and each group has a representative member. Peers communicate and exchange services (information or resources) with the other peers in their groups or out of their groups as shown in Figure 3. In this Figure, R_1, R_2, and R_i are representatives of groups G_1, G_2, and G_i, respectively.

After receiving a service, each peer expresses the value of its trust to the sender with a number based on the quality of the received service. The goal of

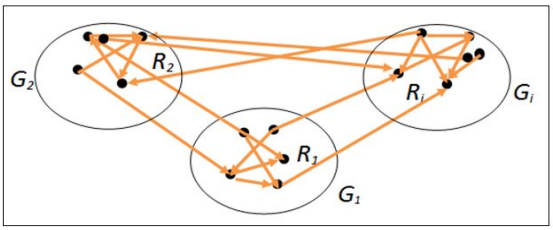

Figure 3. Exchanging of resources or services in *GGRA* network

GGRA algorithm is to achieve reputation of all peers based on the trust values in a distributed, accurate and scalable manner. The algorithm works based on the following assumptions:

(1) Groups in P2P network are established before the aggregation starts. Groups may be formed based on different criteria such as friendship, interest to different subjects or any other similarity measure.
(2) For each group there is a representative (albeit the algorithm can be extended to deal with more than one representative). This peer can be selected from the most reputed or pre-trusted peers. Usually designers and those who are initially connected to P2P networks are not willing to be malicious [24, 26]. Therefore, these nodes are known as pre-trusted and most reputed nodes in the network.
(3) Before the reputation aggregation start , each peer knows its group, its group representative, and all the members of the existing groups.

In addition to the mentioned assumption of the *GGRA* algorithm, the following points should be considered too:

(1) *Joining to the network:* In P2P networks, each node for joining to the network needs to know the addresses of peers who are previously connected to the network. In the proposed algorithm, peers can refer to the bootstrapping site for getting information of groups. This information consists of address of the representative peer and other peers of its own group.
(2) *Grouping Methods:* There are different ways for grouping peers. One way is clustering through the peers' similarity like the interests of them. In this kind of grouping, the group in which peer has the most activity in it, can be chosen as its group. The other way is to use an algorithm which increases efficiency such as creating groups with equal members. However, the best way is to divide peers according to their connectivity graph. This graph is different from the graph which is established for joining, connecting to the network and searching for resources. In this method, pre-trusted nodes or bootstrapping servers use the history of peers that are involved in transactions in order to group according to the connectivity graph. For example social connectivity network can be used as an infrastructure (Connectivity graph). In these networks, Community detection algorithms [27] can be used to divide peers according to the social friendship graph.
(3) *Network type:* In *GGRA*, gossiping and grouping are used for computing the reputation. Gossip based algorithms are distributed algorithms and representatives of groups in *GGRA* work as semi-central agents. Although *GGRA* algorithm takes advantage of a semi-central management scheme, it can be used in distributed, central and semi-central P2P networks, too. In the semi-central networks such as the second generation of P2P networks [11], each server can be considered as representative of each group. In the centralized networks, grouping can be done using the central server, and in distributed networks grouping can be done using the bootstrapping server.

3.2 Execution Phases

GGRA is depicted in Algorithm 1. The algorithm consists of three phases: In the first phase, two gossip algorithms are executed simultaneously, but separately in each group. By the first algorithm, each node calculates the reputation of peers in its own group with *GossipTrust*. Reputation of each group (group as whole not for each member of the group) is calculated through the second algorithm. In Figure 4, the concurrent execution of the gossip algorithm in each group is shown. In the second phase, representatives

Algorithm 1 - *GGRA* Algorithm

▷ **First phase**- Do reputation aggregation in each group
1: Do *GossipTrust* algorithm in groups to get each members' reputation
2: Do the Gossip averaging algorithm 2 to get the reputation of groups
▷ **Second phase**- Do reputation aggregation between groups
3: Do *GossipTrust* algorithm between groups' representatives to get reputation of each group
▷ **Third phase**- Broadcast reputation by aggregation in each group
4: Broadcast reputation of all groups and their members by gossiping in each group

of groups share the reputation values with each other using *GossipTrust* algorithm as shown in Figure 5. Representatives also broadcast the reputation values

of members of their own groups to the other representatives. Finally, in the third phase of *GGRA*, each representative sends the reputation of groups and reputation of other groups' members (which are received from other representatives in the second phase) to the members of its own group. These values are broadcasted in each group via gossiping.

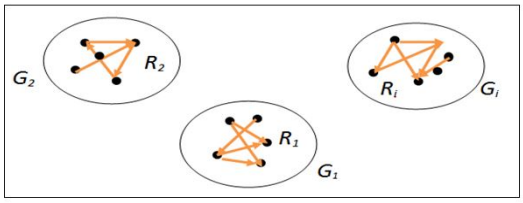

Figure 4. Intra group reputation aggregation

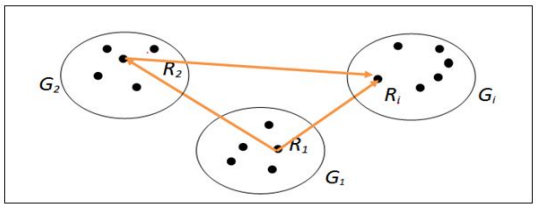

Figure 5. Inter group reputation aggregation

The three mentioned phases are explained in more detail in the following sections:

3.2.1 First Phase

In order to calculate reputation of peers in each group, *GossipTrust* algorithm is executed in the group and between group members. According to the *GossipTrust* algorithm, reputation vector of each node i about the other nodes of group is achieved through Equation 8:

$$\text{if } (i \in G_i, j \in G_j \Rightarrow G_i = G_j) \quad R_i \approx (S^T)^d \times R(0) \tag{8}$$

This equation shows reputation vector of the node i for those nodes that are in the same group. Final reputation value for each member j in R_i is the weighted sum of all trust values about j. Since initially there is no information about the reputation of peers, each member of $R(0)$ in the initial calculation cycle is equal to $1/m$. Other used variables are explained in Table 1. Approximated result in Equation 8 is due to the probable execution of the gossip algorithm. Note that for calculating reputation of members, pre-trusted nodes are also considered as it is shown in Equation 7.

In the first phase, Algorithm 2 is executed for calculating reputation of the groups in each group. In this algorithm, first, each peer i considers the average of trust values to the members of each group as the trust value to that group. Then, i normalizes each group trust between all trust values to different groups. In this way, trust to each group is calculated based on the opinion of each peer i. Then, these values are shared with gossip and averaged between all members of the group using Algorithm 2. Note that Algorithm 2 is executed simultaneously with the first algorithm.

Algorithm 2 - Reputation aggregation of groups in each group

1: **INPUT:** trust matrix R' for all groups, gossip error threshold ε
2: **OUTPUT:** members' reputation matrix R'' for all groups
3: **for all** $z = 1, ..., g$ **do** ▷ All groups aggregate reputations simultaneously
4: **for all** $i \in G_z$ **do** ▷ All members of group z aggregate reputation of other groups simultaneously
5: $\forall j,\ x_{ij} \longleftarrow R'_{ij}$
6: $\forall j,\ w_{ij} \longleftarrow 1$
7: $\forall j,\ R''_{ij}(0) \longleftarrow x_{ij}/w_{ij}$ ▷ Initialize each group reputation
8: $k_i \longleftarrow 0$ ▷ gossip steps counter
9: **repeat** ▷ do gossip steps
10: send $\frac{1}{2}\ R''_i(k_i)$ to a random node q and i itself
11: $\forall j,\ x_{ij} \longleftarrow \sum_{i'}^{r} x_{i'j}\ ,\ w_{ij} \longleftarrow \sum_{i'}^{r} w_{i'j}$ ▷ Aggregate received gossiped values in this step
12: $k_i \longleftarrow k_i + 1$
13: $\forall j,\ R''_{ij}(k_i) \longleftarrow x_{ij}/w_{ij}$ ▷ Update this step reputation
14: **until** $\forall_j, \mid R''_{ij}(k_i) - R''_{ij}(k_i - 1) \mid < \varepsilon$ ▷ Gossip steps finish when all gossiped reputations converge
15: **end for**
16: **end for**
17: **output:** R''

In Algorithm 2, R' is the trust matrix of each member to different groups. Each group has its own R'. R'' is the reputation matrix of the groups which every member obtains.

$$R'_{ij} = \frac{1}{|G_j|} \sum_{k \in G_j} s_{ik} \quad k \in G_j, i \in G_i \tag{9}$$

In Equation 9, R'_{ij} is the trust of i to the group G_j. Note that execution of Algorithm 2 occurs with the main *GossipTrust* aggregation (members' reputation aggregation in the group) concurrently, but only in one cycle. If G_i is considered as one entity and G_j as another entity, R''_{ij} can be declared as the trust of G_i to G_j.

$$R''_{ij} = \frac{1}{|G_i|} \sum_{k \in G_i} R'_{kj} \quad k \in G_i \tag{10}$$

Equation 10 shows calculation of R'' by each peer in the network which leads to the reputation matrix.

Figure 6 shows the level of obtained reputation by the first algorithm in the first phase and Figure 7 shows the level of obtained reputation by the second algorithm.

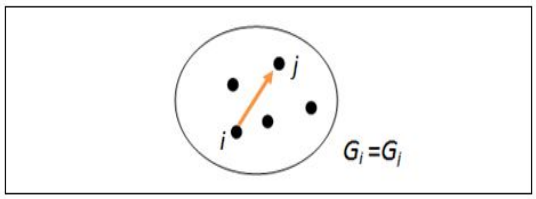

Figure 6. Reputation of peers in the same group

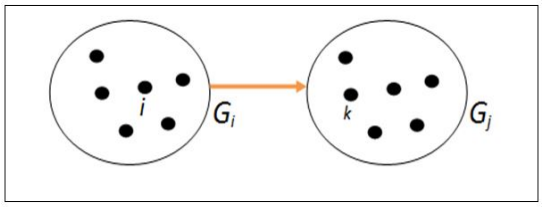

Figure 7. Trust of G_i to G_j

3.2.2 Second Phase

The algorithm used for computing the inter group reputation is again the *GossipTrust* algorithm except that there is no power nodes between different groups and α is zero.

$$groupRep_i \approx (RR''^T)^{d'} \times groupRep(0) \qquad (11)$$

In Equation 11 RR'' is the representatives trust matrix. RR''_{ij} for each representative i is R''_{ij} which is obtained in previous phase in its own group about G_j. $groupRep_i$ in Equation 11 is the final reputation of groups obtained by the representative of the group i. d' is the number of required cycles for convergence of inter group reputation. In the initial calculation cycle, $groupRep(0)$ which is the reputation of groups is equal to $1/g$.

To broadcast the reputation of members by their representatives to other representatives, gossiping is used. Broadcasting occurs concurrently with main reputation aggregation algorithm, but only in the first cycle. Actually in this phase, two types of information are shared; one is for aggregating reputation and the other for broadcasting.

3.2.3 Third Phase

To broadcast other groups' reputation information in each group, a gossip based information dissemination algorithm like one proposed in [1] is used. Broadcasting starts from the representative of the group and in each step each member chooses a random neighbor member and sends the received reputation information to it. In each group, each member can obtain average steps required for execution of one gossip cycle. Therefore, broadcasting is only executed for one cycle.

$$\begin{aligned} &\text{if } (i \in G_i, j \in G_j \Rightarrow G_i \neq G_j) \\ &\quad memberRep_{ij} = groupRep_{G_iG_j} * R_{G_jj} \\ &\text{if } (i \in G_i, j \in G_j \Rightarrow G_i = G_j) \\ &\quad memberRep_{ij} = R_{ij} \end{aligned} \qquad (12)$$

In Equation 12, the first $memberRep_{ij}$ is the reputation of node j by node i out of the group G_j. It is calculated by multiplying R_{G_jj} (the aggregated opinions of group j in the first phase) by $groupRep_{G_iG_j}$ (the calculated reputation of group j in the second phase) (Figure 8). If i and j are in the same group, then there is no need to consider $groupRep_{G_iG_j}$ and $memberRep_{ij}$ is only R_{ij}.

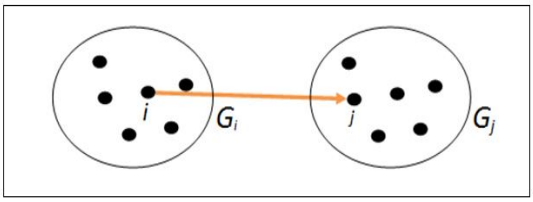

Figure 8. Reputation of j in G_j obtained by i in G_i

Briefly, for calculating the reputation of peers, first in each group, the reputation of all members and all groups are calculated. Then, representatives of groups share and calculate the reputation of groups and finally reputation of members of other groups are broadcasted.

3.3 Time Complexity

In order to calculate the time complexity of the proposed algorithm, the time for executing one step is considered as the time unit. Therefore, the number of steps is the time complexity in *GGRA*. In Equation 13, $T_{GossipTrust}$ is the time complexity of *GossipTrust*:

$$\begin{aligned} &d_1 = O(\log(n)) \\ &T_{GossipTrust} = O(d_1 \log(n)) = O(\log^2(n)) \end{aligned} \qquad (13)$$

Where d_1 is the number of required cycles for aggregating reputation, $O(log(n))$ is the time required for aggregating gossip data in each cycle and n is the number of nodes in the network [3].

Execution time of the *GGRA* algorithm (T_{GGRA}) is the sum of some factors: 1) the execution time required for aggregating reputation data of the members in each

group (first phase), 2) the time needed for aggregating reputation data of groups between representatives (second phase), and 3) the time needed for broadcasting final aggregated data in each group (third phase). Therefore, to calculate T_{GGRA}, at first the time complexity of each phase is analyzed and then summation of them is computed according to Equation 14:

$$T_{GGRA} = T_{FirstPhase} + T_{SecondPhase} + T_{TirdPhase} \quad (14)$$

3.3.1 First Phase

According to the Algorithm 1 (lines 1 and 2), the required time for executing algorithm in each group ($T_{FirstPhase}$) is equal to $O(d_2 \log(m))$. Here, $O(\log(m))$ is the required time for aggregating gossip data in each cycle and d_2 is equal to the number of required cycles for aggregating reputation data. The value of d_2 is different for each trust matrices S and is $O(\log(m))$[3]. As the aggregating reputation of groups in the first phase occurs concurrently with aggregating reputation of members of the group, the time needed for aggregating reputation of groups is not considered.

3.3.2 Second Phase

The required time for executing algorithm between representatives of groups ($T_{SecondPhase}$) based on the 3^{rd} line of Algorithm 1 is $O(d'_2 \log(g))$. In this relation, g is the number of groups, $O(\log(g))$ is the required time for aggregating gossip data in each cycle between representatives and d'_2 is the number of all required cycles for aggregating reputation data. In this way, d'_2 is dependent on reputation matrices R'' according to Equation 10 which is the reputation of groups obtained by members and its complexity is $O(\log(g))$. In the second phase, time for broadcasting reputation of members to representatives is not considered, because it is covered by one cycle execution time.

3.3.3 Third Phase

Broadcasting in the third phase of the algorithm ($T_{ThirdPhase}$)according to the 4^{th} line of Algorithm 1 is $O(\log(m))$.

Since all groups execute the algorithms in the first and third phases concurrently, execution time of the largest group is considered as the required time for aggregating in all groups. So the time complexity of *GGRA* algorithm is calculated in Equation 15:

$$\begin{aligned} d_2 &= O(\log(m)) = O(\log(n)) \\ d'_2 &= O(\log(g)) = O(\log(n)) \\ T_{GGRA} &= O(d_2 \log(m)) + O(d'_2 \log(g)) + O(\log(m)) \\ &= O(d_2 \log(n)) + O(d'_2 \log(n)) + O(\log(n)) \\ &= O(\log^2 n) \end{aligned} \quad (15)$$

Comparison of Equation 13 and Equation 15 shows that time complexity of *GGRA* algorithm is asymptotically the same as *GossipTrust.* However, it is clear that the execution time of *GossipTrust* has direct relation with the number of participating nodes. But in *GGRA,* this time decreases due to division of nodes in groups.

3.4 Number of Exchanged Messages

$M_{GossipTrust}$ in Equation 16 determines the number of exchanged messages in the network using *GossipTrust* algorithm:

$$M_{GossipTrust} = O(d_1 n \log(n)) = O(n \log^2(n)) \quad (16)$$

In each gossip step, each node sends one message. Then, total of n messages are sent in one step. As it is described in the previous section, since there are totally $O(d_1 \log(n))$ steps, the resulting traffic of the messages is of $O(d_1 n \log(n))$. M_{GGRA} in Equation 17 determines the number of exchanged messages in the network by using *GGRA* method:

$$\begin{aligned} d_2 &= O(\log(m)) = O(\log(n)) \\ d'_2 &= O(\log(g)) = O(\log(n)) \\ M_{GGRA} &= O(d_2\, n \log(m)) + O(d'_2\, g \log(g)) \\ &\quad + O(n \log(m)) \\ M_{GGRA} &= O(n \log^2 n) \end{aligned} \quad (17)$$

The number of messages in this method is the sum of the number of messages in intra group aggregation (first phase), inter group aggregation (second phase) and the final broadcast phases (third phase). Details of Equation 17 are explained in following.

3.4.1 First Phase

The maximum required time in each group is $O(d_2 \log(m))$. As said heretofore, the number of messages is proportional to the number of nodes in each step, because each node sends one message. Then, the maximum number of messages in each group is $O(d_2 m \log(m))$. Totally, in the first phase, $O(d_2 n \log(m))$ messages are exchanged in the entire network.

3.4.2 Second Phase

Similar to the explanation of the first phase and according to the time complexity of this phase, the number of exchanged messages between representatives of groups is of $O(d'_2 g \log(g))$.

3.4.3 Third Phase

To broadcast the final reputation data between members of each group, in each step m messages in each group and totally $O(n \log(m))$ messages are exchanged. According to the analysis, the number of messages in both *GossipTrust* and *GGRA* are asymptotically the same.

3.5 Practical Points

There are some practical points which must be considered in the reputation systems and especially in the proposed algorithm:

3.5.1 Churn

One of the problems in P2P networks is churn, i.e. high and dynamic joining or leaving rate of peers in the network. In *GGRA*, peers may do malicious actions and easily leave their group and join another group. Since peers who have communicated with malicious peers are in the previous group, malicious peers are less likely to be recognized in new group. Further, malicious nodes can easily communicate with the nodes outside the group so that the required data for reputation calculation are missed out. To solve this problem, group selection should not be undertaken by the peers themselves. Then, malicious peers could not change reputation results intelligently. The best grouping method is one that divides peers based on communication graph of the peers.

3.5.2 Execution Time of the Reputation Computation

Reputation computation algorithm is executed periodically. It means that in each time period, reputation of all peers is calculated once. A problem occurs when a new peer joins to the network in the time between two consecutive executions and it needs to know the reputation of the other peers. One solution is to ask about the reputation of the target peers from the representatives of the groups which target peers belong to. In order to prevent DoS [1] attacks against representatives, any representative can delegate its responsibility to other peers who have high reputation. Another solution is to reduce the time between executions. The time between two consequent reputation computations depends on the churn rate. As this rate increases, more new members join to the network. Since they were not in the time of reputation calculation, they are not aware of others' reputation. Whatever the rate of requests for attending new members increases, the time between executions of two consequents should decrease.

3.5.3 Storage Space and Traffic of Initial Messages

The process of discovering addresses of all members of the network and the groups' information is a time consuming operation. It needs high volume for storage and produces high traffic. In the networks which malicious nodes are distributed equally in all groups, all groups have partially the same reputation. Therefore, there is no need to know the addresses of members of all other groups and hence to compute the reputation of groups. The problem of not computing reputation of groups is that is that collusive groups can be formed easily. These groups can be formed due to the ability of the nodes that are free to select their groups. To prevent this, reputation of groups should be calculated asynchronously (i.e. in different time periods). In this way the required initial storage space and also the traffic can be reduced.

3.5.4 Membership in More Than One Group

In groups which are overlapped with each other, each member can have a special ID in each group which belongs to. The members evaluate each other according to their specific groups separately. This method is suitable for reputation evaluation of peers based on the context of resources and services. Other possible solution is that each member can only have one ID for all of its groups. In all groups members participate concurrently in reputation calculation.

4 Evaluation

GGRA is mainly practical in networks whose communication pattern is based on intra group relationship or groups are formed based on social relationships i.e. grouping is done based on community graph. Therefore, *Blogcatalog* social network dataset [28] is used as the communication graph for evaluation of *GGRA*. This dataset which consists of 10312 nodes has been extracted from *Blogcatalog* website [2] that is for managing blogs and bloggers. Also in this evaluation, *Gephi* software [29] is used for group detection in *Blogcatalog* friendship graph which is a widely used social network analysis tool. *Gephi* discovers pattern of large graphs

[1] Denial Of Service

[2] www.blogcatalog.com

with different algorithms and different measurements. Implementation of *GGRA* algorithm has been done using *Matlab*. Different samples with different number of nodes have gotten from *Blogcatalog* dataset using *MHRW* algorithm [30]. *MHRW* is a suitable random walk algorithm which is used for sampling from social networks. Random walk sampling starts with a node in the graph and continues with choosing one of its neighbours by using a probability distribution. Then, Groups in samples are recognized using *Gephi*. *Gephi* uses the community detection algorithm which is suggested by *Blondel et al.* [31].

4.1 Simulation Assumptions

In this evaluation, social network infrastructure is used, so in the formed P2P network, the shared resources are information, news or opinions. Good or malicious nodes are determined through reliable and valid shared information.

In the simulated communication pattern of *Blogcatalog* social network, each peer chooses some of its friends randomly and receives information from them. This information can be correct or incorrect according to maliciousness or goodness of the friend. Each peer expresses its trust value to its friend. If the number of correct received information from node j is c and the number of incorrect information is w, then the trust value of node i to node j (r_{ij}) is the difference between correct information c_{ij} and wrong information w_{ij} according to Equation 18 as shown below:

$$r_{ij} = c_{ij} - w_{ij} \tag{18}$$

Then, each peer normalizes all of the positive local trust values to others as explained in Equation 1. Simulated P2P network contains three types of nodes including good, malicious and power nodes. Good nodes send correct news, information, opinions or beliefs. Malicious nodes in addition to sending incorrect information, express contrariwise and wrong trust values i.e. low trust to good nodes and high trust to malicious ones. Other nodes are power nodes which are also good nodes.

In each step of the reputation aggregation, all nodes execute the gossip operation together. In other words, all nodes continue sending and receiving gossip, till all of the nodes have converged. This process happens at each cycle. There are some simulation assumptions as follows:

(1) (a) The information receiver recognizes correctness or incorrectness of the information itself.
    (b) Malicious peers always perform badly and spread incorrect information. Good nodes always spread correct information in contrast.
(2) Each peer only receives information from its friends.
(3) Each peer keeps its friendship and continues receiving information in spite of its friends' attitudes.

4.2 Simulation Results

Simulation variables are commented in Table 2. According to the proposed approach, after some cycles of receiving information from friends (information cycles), reputation is calculated using *GGRA* and local trust values.

In the rest of this section, the practical result of evaluations is described to support the mentioned assertions.

Table 2. Simulation settings

Variables	Description	Default values
n	Number of nodes in network	5000
α	Power nodes reputation increase coefficient	0.15
ϵ	Gossip error threshold	10^{-4}
δ	Reputation aggregation error threshold	10^{-5}
g	Number of groups	33
	Percent of power nodes	1%
	Percent of malicious nodes	50%
	Number of information cycles	100
	Average of received information in each cycle	20

4.2.1 Convergence Speed

In this simulation, convergence speed of *GossipTrust* and *GGRA* algorithms with different number of nodes have been compared. Number of steps needed for execution of *GGRA* algorithm (T_{GGRA}) is the sum of intra group, inter group and broadcast steps as shown in Equation 14. T_{intra} is the maximum execution time (number of steps) between all groups in the first phase, T_{inter} is the execution time (number of steps) between representatives in the second phase and $T_{broadcast}$ is the maximum execution time (number of steps) between all groups in the third phase. To compare the execution time between *GGRA* and *GossipTrust* algorithm, at first *GossipTrust* is performed in the formed network without considering groups. In *GossipTrust* simulation, communication of each peer is only with its friends i.e. each peer sends gossip information only

to its friends. The other variables of simulation are described in Table 2. As the results of evaluation in Figure 9 confirms, there is remarkable decrement in execution time of *GGRA* in comparison with *GossipTrust*. Indeed, due to partitioning peers into different groups, number of participating nodes in *GGRA* algorithm decreases. It is clear that with reduction in number of nodes, the convergence speed increases, too.

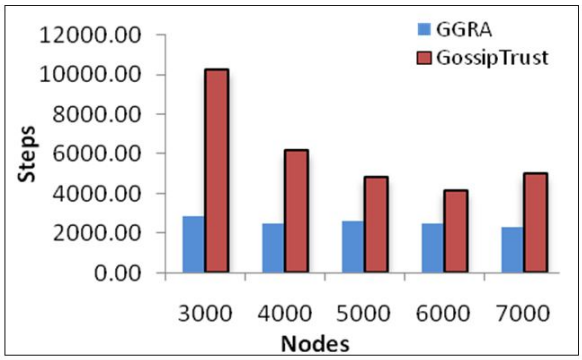

Figure 9. Convergence speed in *GGRA* and *GossipTrust*

Discovering of all intra group members before aggregation start is the other factor in aggregation speedup. In the other words, determination of all members causes to choose neighbors with uniform probability. Therefore, the underlying network is considered as a fully connected graph. Whereas, in *GossipTrust*, peer selection occurs between each peers' friends and it leads to a weakly connected graph. As described heretofore, *GossipTrust* uses the *Kempe* method for gossiping. *Kempe* method is suitable for networks which have strong graphs [6, 7], while *GossipTrust* graph is based on power-law model [26] and it is a weakly connected graph. As much as the graph is weakly connected, convergence occurs later and sometimes convergence never occurs. Further, *Chierichetti et al.* proved that time complexity of the convergence speed in power-law networks in each cycle is $O(log^2 n)$ [7, 32]. According to this explanation, it is obvious that convergence in *GossipTrust* occurs slower than *GGRA*. This is because *GGRA* at every cycle occurs in full graphs while *GossipTrust* occurs in power-law graphs.

4.2.2 Comparison of the Number of Messages

To compare the number of transmitted messages between *GGRA* and *GossipTrust,* Equation 19 is used:

$$M_{GGRA} = n * T_{intra} + g * T_{inter} + n * T_{broadcast}$$
$$M_{GossipTrust} = n * T_{GossipTrust} \tag{19}$$

In this equation, M_{GGRA} is the number of exchanged messages in *GGRA* and $M_{GossipTrust}$ is the number of messages in *GossipTrust.* Equation 19 is driven from Equation 16, 17. Here, the maximum execution time between all groups i.e. T_{intra} is considered for all groups.

Figure 10 shows the number of messages in both algorithms which proves a remarkable decrement in *GGRA* toward *GossipTrust.* As it is explained before, this reduction is achieved through reduction in number of nodes as well as changing the loosely connected network into a strongly connected one.

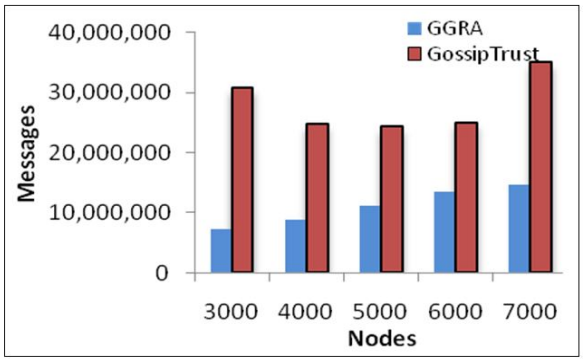

Figure 10. Number of exchanged messages in *GGRA* and *GossipTrust*

4.2.3 Rate of Intragroup Communication

In social groups, it is acclaimed that the most communication is with intragroup members. In this section, the rate of peers' communication with other peers in their own group is measured. The average and the minimum values for a graph with 5000 nodes and with different grouping architecture (33, 17 and 73 groups) are measured. The results of this evaluation, as shown in Table 3, indicate that more than 50% of communications occur within groups.

Table 3. Rate of intragroup communication

Modularity	NumberOf Groups	Minimum Communication	Average Communication
0.75	17	0.62	0.65
0.49	33	0.55	0.63
0.23	73	0.50	0.61

As the intra communication increases, the need for communication with the out of group nodes decreases respectively. The reason is that trust values about intra group peers are also in the groups that peers are belonged to them. Therefore, for computing the reputation value, only trust values of intra group peers are

enough. By using *GGRA* algorithm, most of the required reputation data are provided within groups. In addition, using groups has many advantages such as spending less cost for message exchanging, less storage volume, and less execution time for getting information from all peers in the network. Furthermore, in the first phase of *GGRA* (intragroup gossip), peers can achieve reputation of other intragroup with remarkable accuracy even without execution of the second and the third phases. Modularity in community detection algorithm [31] shows how much groups are determined strongly. As the community detection algorithm works stronger, communication of peers with the out of group nodes decreases and *GGRA* algorithm works better (Table 3).

4.2.4 Comparison of the Gossiped Results with Real Values

Existence of a little difference between the calculated reputation values through gossip algorithm and the actual values is normal due to the stochastic nature of the gossip algorithm. As this difference is less, the algorithm becomes more reliable. In this section, the average difference between calculated reputation values achieved from simulation of *GGRA* algorithm and *GossipTrust* with their corresponding actual values are compared together. The results for a network with 5000 nodes and 33 groups are shown in Table 4. It shows that accuracy of gossiped values in *GGRA* algorithm is more than *GossipTrust*. This is because P2P network in *GossipTrust* are not necessarily strongly connected and nodes are connected with preferential attachment pattern and power-law is governed in these graphs.

Table 4. Mean difference of gossiped results with real values in each group in *GGRA* and mean difference of gossiped results with real values in the whole network in *GossipTrust*

GossipTrust	3.66E-06
GGRA	1.21E-06

4.2.5 Comparison of Malicious Nodes Reputation in GGRA and GossipTrust

Reputation value of malicious nodes in P2P network is an important factor to measure reliability of that network. As the reputation of malicious nodes is less, reputation system works better. The mean reputation value of malicious nodes toward various percentages of them in both algorithms is shown in Figure 11. It shows that average reputation of malicious nodes in *GGRA* is less than *GossipTrust*. In this way, malicious nodes can be recognized from their low reputation. To prove this assertion, a criterion for choosing and determining suitable resources (information) has been considered which acts based on peers' reputation. This criterion is the median value. Simply, it can be declared that the nodes which their reputation are lower than the median value are more likely to be malicious and those with higher reputation are good nodes. Percentage of malicious nodes which their reputation is lower than the median value is computed on both *GossipTrust* and *GGRA* toward various numbers of malicious nodes. The results are demonstrated in Figure 12. It shows that the number of malicious nodes which are lower than the median value in *GGRA* is less than *GossipTrust*. Furthermore, the results indicate that in *GGRA* the good nodes and the malicious nodes both have low reputation. It means that by decreasing the reputation of malicious nodes, the reputation of good nodes decreases, too. Therefore, low reputation cannot show the maliciousness of a node and this is not desired. To analyze the reason of this result in *GGRA*, another evaluation is done. In this one, reputation relationship with the rate of communications with out of group nodes has been measured. In a P2P networks, some of friends are in different groups. This leads to certain transactions between peers both within the groups and out of the groups.

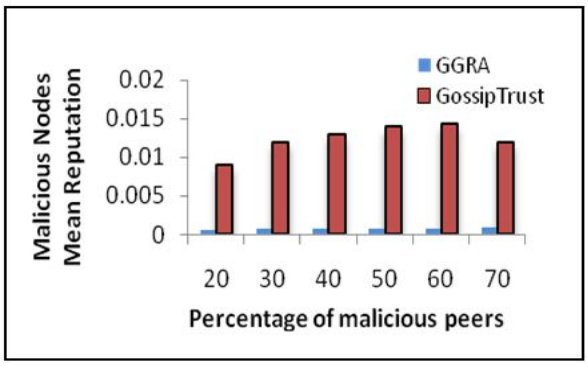

Figure 11. Malicious nodes mean reputation in *GGRA* and *GossipTrust*

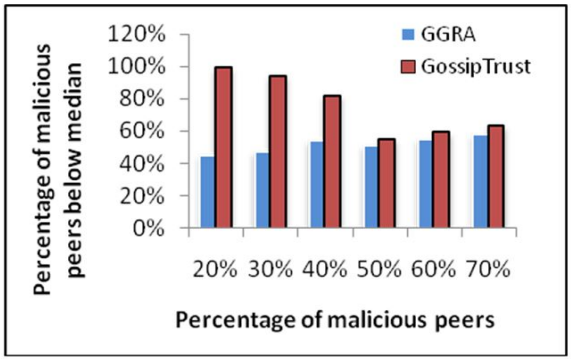

Figure 12. Percentage of malicious peers below median with different Percentage of malicious peers in *GGRA* and *GossipTrust*

In this evaluation, rate of communicating with out of the group has decreased from 100% to 75%, 50%, 25% and 0%. Whatever outer group communication decreases, *GGRA* result becomes closer to *GossipTrust* result and reputation becomes more precise. In the case that all the communications are within the group, the percentage of malicious nodes which are less than the median is equal in both algorithms (Figure 13). Because, communicating with, out of group makes the trust data out of group be missed out in *GGRA* reputation calculation. When the trust data are missed out, reputation of power nodes increases irregularly and reputation of the normal and the malicious nodes, both decreases. Therefore, when trust data are missed out, reputation result are less precise.

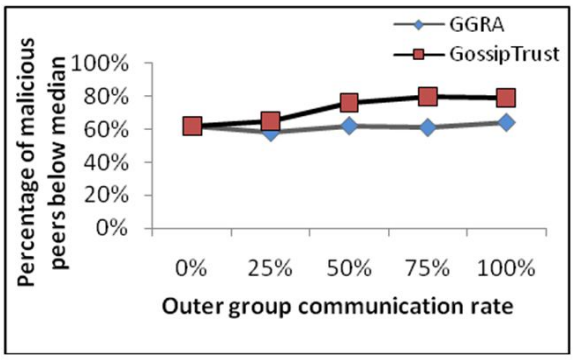

Figure 13. Percentage of malicious peers below median with different outer group communication rate in *GGRA* and *GossipTrust*

5 Conclusion

Malicious behaviours are one of the most challenging issues in P2P networks. Reputation system is known as an acceptable solution for the mentioned problem. According to the particular functionality of the reputation system in this environment, a desired algorithm should have scalability and accuracy features to enrich this type of networks.

In this research, an efficient algorithm, *GGRA*, for reputation aggregation in P2P networks is proposed. In *GGRA*, *GossipTrust* is executed simultaneously in all groups and in a strongly connected graph instead of power-law graph. Therefore, reduction in time, number of messages and storage space are achieved as advantages.

According to the evaluation results, the proposed *GGRA* algorithm is useful in groups which most of communications are with intra group peers. This property is seen in social based groups which itself helps P2P systems to be survival. This is because the peers participate more in these systems, they try to defend from their groups and refuse showing malicious actions. Using social groups eliminates the need for knowing reputation of all peers. Therefore, lots of traffic and time can be saved. These advantages make using social networks as an infrastructure for P2P networks to be plausible and more efficient. Furthermore, using semi-central representatives in the proposed model helps to reduce computations and increase efficiency as much as possible.

References

[1] Xuemin Shen, Heather Yu, John Buford, and Mursalin Akon. *Handbook of peer-to-peer networking*, volume 1. Springer Heidelberg, 2010.

[2] Kannan Govindan and Prasant Mohapatra. Trust computations and trust dynamics in mobile adhoc networks: a survey. *Communications Surveys and Tutorials, IEEE*, 14(2):279–298, 2012.

[3] Runfang Zhou, Kai Hwang, and Min Cai. Gossiptrust for fast reputation aggregation in peer-to-peer networks. *Knowledge and Data Engineering, IEEE Transactions on*, 20(9):1282–1295, 2008.

[4] Yoram Bachrach, Ariel Parnes, Ariel D Procaccia, and Jeffrey S Rosenschein. Gossip-based aggregation of trust in decentralized reputation systems. *Autonomous Agents and Multi-Agent Systems*, 19(2):153–172, 2009.

[5] Ruchir Gupta and Yatindra Nath Singh. Trust estimation and aggregation in peer-to-peer network using differential gossip algorithm. *CoRR*, abs/1210.4301, 2012.

[6] Mark Jelasity, Alberto Montresor, and Ozalp Babaoglu. Gossip-based aggregation in large dynamic networks. *ACM Transactions on Computer Systems (TOCS)*, 23(3):219–252, 2005.

[7] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-based computation of aggregate information. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 482–491. IEEE.

[8] Masanori Yasutomi, Yo Mashimo, and Hiroshi Shigeno. Grat: Group reputation aggregation trust for unstructured peer-to-peer networks. In *Distributed Computing Systems Workshops (ICDCSW), 2010 IEEE 30th International Conference on*, pages 126–133. IEEE.

[9] Eyuphan Bulut and Boleslaw K Szymanski. Constructing limited scale-free topologiesover peer-to-peer networks. *Parallel and Distributed Systems, IEEE Transactions on*, 25(4):919–928, 2014.

[10] Matei Ripeanu, Adriana Iamnitchi, and Ian Foster. Mapping the gnutella network. *IEEE Internet Computing*, 6(1):50–57, 2002.

[11] Klaus Wehrle and Ralf Steinmetz. *Peer-to-Peer systems and applications*. Springer, 2005.

[12] Jian Yang, Yiping Zhong, and Shiyong Zhang. An efficient interest-group based search mechanism in unstructured peer-to-peer networks. In

Computer Networks and Mobile Computing, 2003. ICCNMC 2003. 2003 International Conference on, pages 247–252. IEEE.

[13] Weiguo Wu, Wenhao Hu, Yongxiang Huang, and Depei Qian. *Group-based Peer-to-Peer Network Routing and Searching Rules*, pages 509–514. Springer, 2005.

[14] Abhilash Gummadi and Jong P Yoon. Modeling group trust for peer-to-peer access control. In *Database and Expert Systems Applications, 2004. Proceedings. 15th International Workshop on*, pages 971–978. IEEE.

[15] Wen Ji, Shoubao Yang, Dong Wei, and Weina Lu. Garm: A group-anonymity reputation model in peer-to-peer system. In *Grid and Cooperative Computing, 2007. GCC 2007. Sixth International Conference on*, pages 481–488. IEEE.

[16] Yong Zhang, Hongliang Zheng, Yining Liu, Keqiu Li, and Wenyu Qu. A grouptrust model based on service similarity evaluation in p2p networks. *International Journal of Intelligent Systems*, 26(1):47–62, 2011.

[17] Mansooreh Ezhei and Behrouz Tork Ladani. Gtrust: A group based trust model. *The ISC International Journal of Information Security*, 5(2):155–169, 2014.

[18] Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys (CSUR)*, 42(1):1, 2009.

[19] Ferry Hendrikx, Kris Bubendorfer, and Ryan Chard. Reputation systems: A survey and taxonomy. *Journal of Parallel and Distributed Computing*, 75:184–197, 2015.

[20] M Srikanth and KB Madhuri. Secure and effective p2p reputation system using trust management and self certified cryptographic exchanges, 2013.

[21] Kazuhiro Kojima. Grouped peer-to-peer networks and self-organization algorithm. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 3, pages 2970–2976. IEEE.

[22] Huirong Tian, Shihong Zou, Wendong Wang, and Shiduan Cheng. A group based reputation system for p2p networks. In *Autonomic and trusted computing*, pages 342–351. Springer, 2006.

[23] Ajay Ravichandran and Jongpil Yoon. Trust management with delegation in grouped peer-to-peer communities. In *Proceedings of the eleventh ACM symposium on Access control models and technologies*, pages 71–80. ACM, 2006.

[24] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, pages 640–651. ACM, 2003.

[25] Asaki Matsumoto, Yo Mashimo, Masanori Yasutomi, and Hiroshi Shigeno. Ilgt: Group reputation aggregation method for unstructured peer-to-peer networks. In *Parallel and Distributed Systems (ICPADS), 2010 IEEE 16th International Conference on*, pages 197–204. IEEE.

[26] Runfang Zhou. *Scalable Reputation Systems for Peer-to-peer Networks*. PhD thesis, 2007.

[27] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[28] Reza Zafarani and Huan Liu. Social computing data repository at asu. *School of Computing, Informatics and Decision Systems Engineering, Arizona State University*, 2009.

[29] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: an open source software for exploring and manipulating networks. In *ICWSM*.

[30] Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. Walking in facebook: A case study of unbiased sampling of osns. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE.

[31] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[32] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. Rumor spreading in social networks. *Automata, Languages and Programming*, pages 375–386, 2009.

Safieh Ghasemi Falavarjani received B.S. and M.S. in Computer Engineering from University of Isfahan, Isfahan, Iran in 2009 and 2013, respectively. Her research interests include software and network security, trust and reputation systems.

Behrouz Tork Ladani received his B.S. in Software Engineering from University of Isfahan, Isfahan, Iran in 1996, and M.S. in Software Engineering from Amir-Kabir University of Technology, Tehran, Iran in 1998. He received his Ph.D. in Computer Engineering from Tarbiat-Modarres University, Tehran, Iran in 2005. He is currently associate professor in Faculty of Computer Engineering at University of Isfahan. Dr. Ladani is member of Iranian Society of Cryptology (ISC). He is also managing editor of the Journal of Computing and Security (JCS) and member of editorial board of the International Journal of Information

Security Science (IJISS). Dr. Ladani's research interests include soft security and computational trust, software security, cryptographic protocols, and formal verification.

Simin Ghasemi was born in Zanjan, Iran in 1987. She received her B.S. in 2010 from Institute for Advanced Studies in Basic Sciences (IASBS), and M.S. from Sharif University of Technology on Data Security in September 2010. She is already a faculty member of Payam Noor University of Zanjan. Her research interests include data and database security, trust based models and database outsourcing.