GroupTrust: Dependable Trust Management

Xinxin Fan, Ling Liu, *Fellow, IEEE*, Mingchu Li, and Zhiyuan Su, *Member, IEEE*

Abstract—As advanced computing and communication technologies penetrate every aspect of our life, we have witnessed the persistent growth of open systems where entities interact with one another without prior knowledge or experiences. Trust becomes an important metric in such open systems. This paper presents a dependable trust management scheme - GroupTrust, and a working system to support GroupTrust. It makes three original contributions. First, we identify a set of vulnerabilities that are common in existing reputation based trust models. We show that reputation trust built solely on direct experiences or by combining direct experiences with uniform trust propagation can be vulnerable. Second, we develop GroupTrust, a dependable trust management scheme to provide reliable trust management in the presence of dishonest ratings, malicious camouflage, and malicious collusive behaviors. The GroupTrust scheme is novel in two aspects: (i) we develop a pairwise similarity based feedback credibility to enhance the resilience of trust computation in the presence of dishonest ratings; (ii) we propose to propagate trust based on a Susceptible-Infected-Recovered (SIR) model, which defines trust propagation threshold to control how trust should be propagated. Finally, we evaluate the effectiveness of GroupTrust against four threat models using both simulated and real world datasets. Our experimental results show that feedback credibility based local trust computation can effectively constrain strategically malicious participants from taking advantages of their dishonest ratings. SIR-based trust propagation control enables safe trust propagation and blocks irrational trust propagation. We show that GroupTrust scheme significantly outperforms other trust models in terms of both performance and attack resilience in the presence of dishonest feedbacks, sparse feedbacks, and strategically malicious participants against four representative threat models.

Index Terms—Trust propagation, trust and reputation management, decentralized computing network, open systems

1 Introduction

Reputation and trust are critical capability for managing services and interactions among networked entities in large-scale open systems [1], ranging from cloud data centers, decentralized computing networks, such as supply-chain management systems, social networks, mobile networks, online communities, scientific or professional networks, and Internet of Things (IoT) applications. A common characteristics shared by these networked computing systems is the fact that entities may interact with one another without prior experience or knowledge in order to obtain services or accomplish certain tasks. Reputation based trust management has been used as an effective service selection criterion in many real world systems, represented by Amazon, eBay, Twitter, to name a few [2], [3].

Although existing reputation based trust models vary in how they establish trust, most of them build trust based on their transactional experiences or direct interactions between participants of a network. Thus, the reputation based trust can be seen as a network-wide trust measure obtained by collecting and integrating the experiences that each participant has had with other members through interactions or transactions [4]. Such trust measurement can help participants to better manage their transactions by encouraging trustworthy behavior, and protecting good participants from dishonest and possibly malicious participants [5], [6].

Trust and reputation management have been an active research area over the last decade. Most of the trust and reputation computational models are developed by utilizing formal methods, such as contextual or similarity based computation [4], [7], fuzzy logic theory [8], Bayesian network [9], subjective logic [10] and social cognitive methods [11]. EigenTrust [5] is one of the most popular reputation based trust models, which utilizes the eigenvector based propagation kernel [5] to compute trust such that the trust of a participant is based on two factors: (i) the feedback ratings of other participants with whom this participant has had transactions in the past, and (ii) the eigenvector of this participant in the rating network of participants, aiming at addressing the problem of sparse networks in terms of direct feedback rating based referrals. The eigenvector based trust propagation [5] enables the trust establishment of a participant by combining direct experiences measured by feedback ratings and indirect experiences obtained through its k-hop neighbors in the rating network, also referred to its circle of "friends" within the k-hop neighborhoods. Although the eigenvector based trust model [5] can scale the trust computation to large-scale networks, it fails to function dependably in the presence of strategically collusive and dishonest participants.

In this paper we present GroupTrust, a dependable trust management scheme, and a working system to support GroupTrust. Our proposed trust protocol is effective with two novel features: (1) feedback credibility weighted local

- *X. Fan was with School of Software Technology, Dalian University of Technology, Dalian 116620, and with the College of Computing, Georgia Institute of Technology, Atlanta, GA, 30332. E-mail: xinxfan@yahoo.com*
- *L. Liu is with the College of Computing,Georgia Institute of Technology, Atlanta, GA, 30332.*
- *M. Li is with the School of Software Technology,Dalian University of Technology, Dalian, 116620, China.*
- *Z. Su is with the State Key Laboratory of High-end Server & Storage Technology, Jinan, China.*

Xinxin Fan performed initial development of this work while he was a visiting PhD student at Georgia Institute of Technology under the CSC scholarship. Ling Liu is partially supported by the National Science Foundation under grants IIS-0905493, CNS-1115375, IIP-1230740, and SaTC 1564097. This work is also supported by National 863 Plan under grant: SS2015AA010302, National Science Foundation under grants: 61272173, 61100194.

Manuscript received Nov. 28, 2015; revised xx xx, 20xx.

(direct) trust and (2) controlled trust propagation kernel based on Susceptible-Infected-Recovered (SIR) model. Concretely, the good participants are grouped together according to the SIR based trust propagation kernel with feedback credibility based aggregate trust as the control knob to constrain the trust of good participants to be propagated only to the group of good participants who have similar trust and feedback rating behavior. Thus, the concept of group of participants with similar trust and feedback rating behavior is essential and critical to the development of the GroupTrust model and protocols. We show that GroupTrust model is dependable for managing reputation and trust in large-scale networks that are sparse because a large number of participants typically only interact with a small number of other participants. By dependable, we mean that (i) the reputation trust of a good participant should remain to be reliable in the presence of dishonest or malicious participants, (ii) new participants should be able to build their reputation over time, and (iii) the reputation trust of a malicious participant should be dropped sharply once he is found to be dishonest or misbehave, for example, by providing inauthentic contents or bad services.

This paper makes three original contributions. First, we identify a set of vulnerabilities that are common in existing reputation based trust models. We show that reputation trust built solely on direct experiences or by combining direct experiences with uniform trust propagation may still be vulnerable. Second, we develop GroupTrust to provide reliable trust management in the presence of dishonest ratings, malicious camouflage, malicious collusive behaviors. GroupTrust is original in two aspects: (i) we develop a pairwise similarity based feedback credibility to enhance the resilience of trust computation in the presence of dishonest ratings; and (ii) we introduce a SIR model based trust propagation control mechanism, which defines trust propagation threshold in order to determine how trust should be propagated. We evaluate the effectiveness of GroupTrust against four threat models using both simulated and real world datasets. We show that our feedback credibility based local trust computation can effectively constrain strategically malicious participants and the impact of their dishonest ratings. Our SIR-based trust propagation control encourages safe trust propagation and blocks irrational trust propagation. We also show that GroupTrust significantly outperforms existing trust models with respect to both performance and attack resilience in the presence of dishonest feedbacks, sparse feedbacks, and strategically malicious participants.

2 Overview and Problem Statement

This section briefly describes the reference trust model and the threat models used to compute the trust and compare different trust models in terms of their strength and weakness in terms of performance and attack resilience.

2.1 Reference Trust Model

In a large-scale trust enabled network system of n participants, participants interact with one another to provide services and consume services with one another. Upon the completion of a transaction between a pair of participants, the consumer of the service will rate the provider of the service in terms of its transaction quality, denoted by $tr(i, j)$. Each participant i in the network can only rate another member j if it has an actual transaction with participant j, such as purchased a product or downloaded a music file from j. The rating scheme can be either binary [5], [7] or multi-scale [12]. For example, with binary rating model, i can give j the positive feedback rating by setting $tr(i, j)$=1 or negative by setting $tr(i, j)$=-1. By default, $tr(i, j)$=0 for i, j =1, ..., n, and it implies that i has never had any transaction with j. We denote this transaction based local trust from participant i to another participant j by s_{ij}, such that $s_{ij} = \sum tr(i, j)$, i.e., the sum of individual feedback ratings. For binary rating scheme, this is equivalent to the difference between satisfactory and unsatisfactory transactions that participant i has received from participant j, namely $s_{ij} = sat(i, j) - unsat(i, j)$. An obvious problem of using s_{ij} as the trust value that i gives to j is the risk of dishonest or malicious raters, namely, dishonest or malicious participants may give arbitrarily high local trust values to malicious participants, and arbitrarily low local trust values to good participants. A common way to alleviate this problem [2], [3], [4], [5], [7], [8] is to use the normalized local trust value that i has over j, denoted by c_{ij}, such that s_{ij} is normalized by the satisfactory score from all the participants in the network with whom i has had direct transactions:

$$\begin{array}{l} c_{ij} = max(s_{ij}, 0)/\sum_k max(s_{ik}, 0) if \sum_k max(s_{ik}, 0) \neq 0 \\ c_{ij} = p_j \ \ otherwise \end{array} \quad (1)$$

P=$\{p_j\}$ is the set of pre-trusted participants, p_j=1/$|P|$ for $j \in P$ and p_j=0 otherwise, serving as the central authority of the system. The size of P is relatively small compared to the network size of n and is a system configuration parameter. The use of pre-trusted participants helps bootstrap the trust system initially [5]. Based on local trust c_{ij} (i, j=1, ..., n), we can construct a local trust network for the n participants such that two participants, i, $j \in [1, n]$ and $i \neq j$, have a direct edge if there is a local trust relationship defined by $c_{ij}(>0)$. For very large n, this trust network may be very sparse: for many participants, it may have transactions with only a few participants, thus c_{ij}=0 for most $j \in [1, n]$. In this scenario, it will be very hard for i to select the right service providers based on its local trust relationship with only a few entities. This motivates the reference model for trust computation should include both local trust computation and trust propagation.

By trust propagation, even if c_{ij}=0, we can establish an indirect trust from participant i to participant j if j is reachable through k-hop graph traversal. We call the set of nodes reachable from i within k-hop the circle of "friends" of i. We can compute the indirect trust c_{ij} by the weighted summation of c_{iq} and c_{qj} in an iterative fashion, i.e., $c_{ij} = \sum_q c_{iq} \cdot c_{qj}$.

Let n denote the total number of participants in the system, we can define C as the matrix $[c_{ij}]$ with n rows and n columns. Let $\vec{t}^{(k+1)}$ denote the global trust vector of size n to be computed at the $(k+1)$th round of iterations, $0<k<n$. We define $\vec{t}^{(k+1)} = (1-a)C^T\vec{t}^k + a\vec{p}$, where a is the random probability that a participant trusts none but pre-trusted participants with probability 0.1 in the network, and $\vec{p}$ denotes the initial trust vector where only pre-trusted

participants have non-zero trust values, and each is set to $1/|P|$. We denote each element of the trust vector $\vec{t}^{(k+1)}$, by $t_i^{(k+1)}$, we transform the above matrix form as follows:

$$t_i^{(k+1)} = (1-\alpha)(c_{1i}t_1^{(k)} + \cdots + c_{ni}t_n^{(k)}) + \alpha p_i \quad (2)$$

This formula shows that the reputation trust of participant i can be computed by aggregating the local trust values that i has received from all other participants in the system.

2.2 Trust-enabled Service Selection

By establishing and maintaining trust evaluation in a networked system, it enables participants to utilize the trust scores to select the right participant as its service provider. Two most commonly adopted trust-based service selection schemes are deterministic and probabilistic methods. The deterministic algorithm always chooses the participant with the highest trust score among those who respond to its service request as the provider of its request. However, this method can easily overload the participant with the highest trust value. The probabilistic algorithm addresses this problem by choosing a participant i as the provider according to the trust based probability, computed by $t_i/\sum_{j=1}^{R} t_j$, ensuring that a participant with higher trust score will have higher probability to be selected as service provider. To overcome the problem of cold start with new members, we augment the above trust based service selection scheme by introducing different refinements. For example, by setting some default probability [5], say 10%, the system can select randomly from those participants whose trust scores are zero as the service provider (e.g., download source). This refined selection method provides some opportunity for new participants to build their reputation trust in the system while preventing the system from overloading those participants with high trust scores.

2.3 Threat Models

One way to evaluate the effectiveness of trust models is to measure its resilience to different attack strategies used by malicious participants. We below describe the four most widely used threat models, introduced initially in [5] and used by many in reputation trust literature [7], [8], [12].

Threat Model A (Independently Malicious). Malicious participants are independent and simply provide bad services, e.g. upload inauthentic files, and dishonest feedback ratings, without colluding with other malicious participants.

Threat Model B (Chain of Malicious Collectives). Malicious participants collude with each other and deterministically give one another high local trust value. This results in a malicious chain with everyone in the chain having mutual high local trust values (say 1.0). Malicious participants always provide bad service (such as inauthentic file) when selected as a service provider (such as a download source).

Threat Model C (Malicious Collectives with Camouflage). Malicious participants provide good services in f% of all cases when selected as service providers in order to obtain high local trust values from good participants and at the same time, these camouflage malicious participants always provide dishonest feedbacks to good participants.

Threat Model D (Malicious Spies). Malicious participants are strategically organized into two groups: the group of malicious spies, called type D participants, who act as good participants in providing services to increase their global trust and use the trust they obtain to boost the trust of another group of malicious collective (type B) participants who always provide bad services and dishonest ratings.

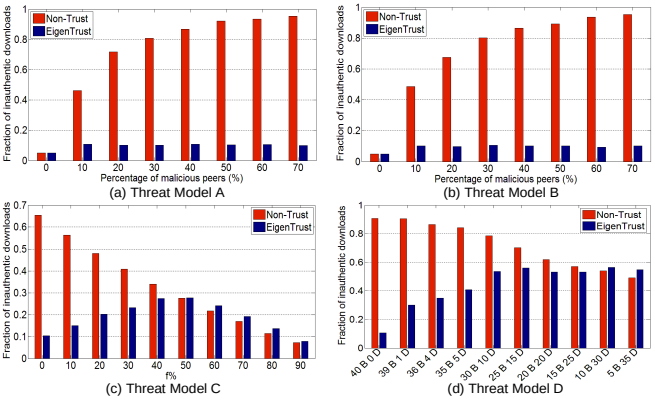

Fig. 1. EigenTrust performance in Threat Models A, B, C and D.

2.4 Problem Definition

To gain a better understanding of the potential vulnerabilities of existing trust models, we evaluate the reference trust model against the above four attack models using the same setup as the one reported in EigenTrust [5], such that the total number of participants in Threat Models A and B are 63 (3 pre-trusted participants), wherein the ratios of malicious participants are set as 0, 10%, 20%, 30%, 40%, 50%, 60% and 70%. For Threat Model C, the disguise, good and pre-trusted participants are 20, 50 and 3 respectively. The evaluation is focused on measuring the resilience of the trust models under varying camouflage percentage f% under a constant ratio of malicious participants at 27%. For Threat Model D, the malicious participants (type B and type D with different combinations), good and pre-trusted participants are 40, 60 and 3 correspondingly. The evaluation of trust models under Threat Model D is performed by varying the ratio of type D and type B malicious participants under a constant ratio of malicious participants at 39%. Fig. 1 shows the results, which are consistent to those reported in EigenTrust [5]. We make three observations: (i) the reference trust model works effectively compared to Non-Trust scenario under Threat Models A and B with up to 70% malicious participants, but it performs poorly against Threat Models C and D even when the number of malicious participants is small, such as about 27% and 39% respectively in Fig. 1(c) and (d). This is because the malicious participants provide no good services, and consequently fail to get positive feedback ratings under Threat Models A and B. (ii) Under Threat Model C, the effectiveness of reference trust model deteriorates very fast as the camouflage percentage f increases. When f is 50% or higher, even with one third of malicious participants, the malicious camouflage participants can gain positive feedback ratings from other participants and get high global trust score while providing dishonest feedback ratings on other participants (bad ratings for good participants and good ratings for malicious participants). Not surprisingly, when f>50%, the reference trust model has high fraction of

bad services, and performs worse than Non-Trust systems. (iii) Similarly, under Threat Model D, as the number of malicious spies (type D) increases by 25% of the total malicious group, the reference trust model starts deteriorating and when the malicious spies are up to 62.5% of the total malicious colluding group, the reference trust model is no longer effective.

In the subsequent sections, we identify two root causes of the above problems from the facets of feedback credibility and trust propagation kernel.

2.4.1 Feedback Credibility

Compared to many existing trust models that establish trust solely based on direct experiences [1], [2], [4], [6], [8], the reference trust model, represented by EigenTrust, can effectively address the problem of sparseness in trust networks. However, as shown in Fig. 1 (c)-(d), the reference trust model is inherently vulnerable when there exist participants that are strategically malicious and dishonest. One root cause of such vulnerability lies in the assumption that good participants always exhibit trustworthy behaviors for both transaction services and feedback ratings, and on the contrary, bad participants are always bad in providing both bad services (e.g., inauthentic contents) and dishonest in providing ratings. Clearly, this assumption is unrealistic and leads the reference trust model and EigenTrust vulnerable to malicious behaviors under Threat Models C and D. We argue that transaction behavior and rating behavior may not always be consistent even for good participants. For instance, a participant may provide good services at all times but may occasionally give dishonest feedback rating about another participant's service due to jealousy or competition related motives, even though it had received satisfactory services from the other participant [4]. On the other hand, malicious participants may strategically provide good services in $f\%$ of all cases when selected as service providers, but are dishonest in feedback ratings by giving bad ratings to all good participants but good ratings to bad participants (Threat Model C). Alternatively, malicious participants may exhibit spy behavior (type D) by providing good services at all times but give dishonest ratings to their malicious collusive (type B) participants (Threat Model D).

2.4.2 Controlled Trust Propagation

Another root cause for the reference trust model being vulnerable under Threat Models C and D is due to its adoption of a uniform eigenvector propagation kernel. By employing the uniform propagation kernel, trust of good participants can be easily manipulated and misused by dishonest participants to raise their trust scores.

Recall Formula (2) in section 2.1, if two participants i and j have no prior transaction experiences, but they are reachable by graph traversal in the rating network, then the reference trust model will be able to establish trust between participants i and j through iterative trust computation. The number of hops in the shortest path from i to j defines the number of iteration rounds we need to compute the trust that participant i has over participant j, denoted by t_{ij}. The eigenvector based trust models utilize power iteration to compute the reputation value for each participant based on the normalized local trust matrix C (recall Formula (2) in section 2.1). A large number of iteration rounds can guarantee the computation stability since it converges to the left principle eigenvector of local trust matrix C.

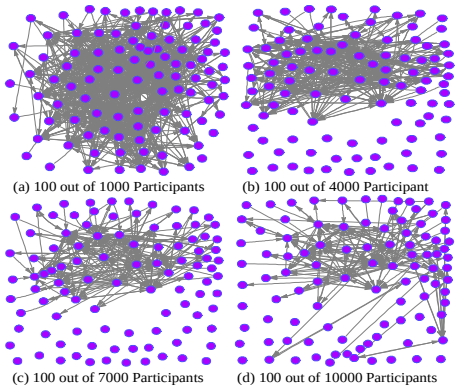

Fig. 2. Density/Distribution of feedback rating network connectivity.

Fig. 2 shows that as the size of the network increases, the rating based local trust network (100 nodes) becomes fairly sparse with more isolated participants in the real world network Epinions [7]. Thus, the systems that rely solely on direct trust using rating based transaction experience will not be effective as many low-degree participants who know little about most members of the service network will not be able to use experience based trust measures to select a service in large-scale sparse networks. One way to build trust among participants with no prior experiences is to use trust propagation. The most widely used trust propagation model is based on uniform propagation kernel, first introduced in EigenTrust [5].

By uniform trust propagation, the trust of a participant is uniformly propagated first to all its neighbors, then to neighbors of its neighbors, eventually to everyone in its neighbor circle of k-hops. This uniform trust propagation model works well when there are no malicious collectives with camouflage or spy behavior, as the number of iteration rounds increases, the trust scores of malicious participants may be increased, and the trust scores of good participants may be manipulated, which may lead to the increase of the fraction of bad services. Consider Threat Model C, with higher percentage of camouflage f, malicious participants can obtain more positive feedback ratings from good participants, gain higher trust, and boost the trust scores of all malicious participants in the chain of trust propagation. As the number of iteration rounds increases, the malicious collectives continue to obtain higher trust scores. At the same time, the trust scores of good participants may drop continuously due to increased amount of dishonest feedback ratings by malicious collectives. Similar observations can be made under Threat Model D. Another issue with trust propagation is the need to determine the right convergence condition (e.g., the number of iterations) such that all or a majority of the participants can receive trust scores via trust propagation when the algorithm converges.

2.4.3 Overview of Solution Approach

We have analyzed and identified the two main root causes that lead to the serious vulnerability of all existing experience based trust models and uniform trust propagation

models, represented by EigenTrust, under Threat Models C and D. We argue that a dependable trust management scheme should be attack resilient in the presence of Threat Models C and D. We propose to mitigate such vulnerabilities by developing GroupTrust with a suite of attack resilient methods with two unique features. First, we argue that a participant's local trust value should be computed by taking into account both its transaction based reputation (rating by experiences) and its feedback credibility. This will enable us to detect malicious participants prior to trust propagation. Concretely, GroupTrust adapts the pairwise feedback rating similarity [4] to define the relative rating credibility of a participant with respect to another participant. If i has higher rating similarity to j, showing that they have similar feedback rating behaviors, then the feedback credibility of participant i with respect to participant j is high. Instead of computing local trust based solely on direct experiences via normalized feedback ratings as shown in Formula (1), GroupTrust weights the normalized rating score from one participant to another with their rating similarity based feedback credibility.

The second novel feature of GroupTrust is to introduce conditional trust propagation to control when and how much the trust score of a participant can be propagated to its neighbors. Concretely, we define the propagation threshold by introducing two control parameters: *transmit probability* and *recovered rate*, which simulate the dynamics in trust behavior. This enables us to control the amount of trust propagating from good participants to bad participants through early detection of malicious behaviors and just-in-time termination of mischievous trust propagation. Although a system-wide trust propagation threshold is straightforward, we argue that it is vulnerable for malicious manipulation. Thus we use a pairwise threshold in GroupTrust. If participant i has high pairwise similarity with participant j, then the transmit probability from participant i to participant j is high, and when this transmission probability is above the recovered rate, then the trust propagation from participant i to participant j is permitted. Otherwise, the trust propagation from participants i to j will be blocked at the current propagation iteration. Furthermore, in GroupTrust we utilize the recovered rate to gain better control not only on when but also how much participant i will propagate its trust to participant j. Our experiments show that GroupTrust powered by these two novel features outperforms existing trust mechanisms and highly resilient against malicious manipulations and attacks.

3 GroupTrust: A Dependable Trust Model

In this section, we present the design of GroupTrust. We enhance the dependability of GroupTrust against attacks in both trust computation and trust propagation process.

3.1 Similarity based Local Trust Computation

In GroupTrust we compute local trust that a participant i places on another participant j in three steps. First, we aggregate satisfactory and unsatisfactory ratings by summarization. Second, we normalize the aggregated rating by incorporating the total amount of transactions between two participants as the denominator. In the third step, we compute the local trust from participant i to participant j by weighting the normalized aggregate rating by their pairwise rating similarity based feedback credibility.

Normalized Rating Aggregation. Computing local trust based solely on the difference between the number of satisfactory transactions and the number of unsatisfactory transactions may suffer from a number of vulnerabilities. Consider the following three scenarios: (1) participant i has 10,000 satisfactory and 9,980 unsatisfactory transactions with j_1; (2) participant i has 100 satisfactory and 80 unsatisfactory transactions with participant j_2; (3) participant i has 20 satisfactory and 0 unsatisfactory transactions with j_3. Clearly by the formula $s_{ij} = sat(i,j) - unsat(i,j)$, we have that $s_{ij_1} = s_{ij_2} = s_{ij_3} = 20$, namely, their local trust values are equal. However, if we look at the percentage of satisfactory transactions, then the perceived local trust of j_2 would be much better than that of j_1 because j_2 has 11% transactions rated satisfactory compared to only 0.1% satisfactory transactions by j_1. In addition, establishing trust by relying on only the percentage of satisfactory transactions may be vulnerable since adversary may easily subvert the system by simply performing a small number of good transactions to gain high trust value, such as j_3. It is also intuitive that j_1 has performed significantly larger number of good transactions than j_3. Thus, the simple sum based rating comparison is unreliable for local trust computation. A root cause for such problems is the lack of consideration on the number of factors that may play a role in the local trust computation, such as the total number of transactions, the percentage of unsatisfactory transactions, to name a few. In addition, GroupTrust uses a system defined error bound θ to reflect that good participants may perform bad with probability θ due to some unintentional reasons, e.g., unreliable data readings resulting from the cooling problem or compromises. In our experiments we set θ=5%. We define the satisfactory rating score that participant i gives to participant j, denoted by s_{ij}, as follows:

$$s_{ij} = \begin{cases} \frac{sat(i,j)}{sat(i,j)+unsat(i,j)+1} & \frac{unsat(i,j)}{sat(i,j)+unsat(i,j)+1} \leq \theta \\ \frac{1}{2} & otherwise \end{cases} \quad (3)$$

Given that participant i may have different satisfactory scores for different participants j and some may have negative values, we further normalize the aggregate rating score s_{ij}, denoted by r_{ij} or $r(i,j)$ by using the sum of the ratings that participant i has received from all other participants as the denominator and ensuring that r_{ij} is nonnegative:

$$r_{ij} = \begin{cases} \frac{max(s_{ij},0)}{\sum_j max(s_{ij},0)} & if \ \sum_j max(s_{ij},0) \neq 0 \\ p_j & otherwise \end{cases} \quad (4)$$

where p_j represents the set P of pre-trusted participants of the network, p_j=1/$|P|$ for $j \in P$ and p_j=0 otherwise.

Feedback Rating Credibility. In GroupTrust we adapt the concept of feedback rating credibility [4] in our local trust computation model. The use of feedback credibility measure is motivated based on three observations: (i) Two good participants may give very similar feedback ratings to the common set of participants with which they have had interactions or transactions in the past. (ii) Two malicious participants, on the other hand, may give very similar

feedback ratings to the common set of (good or malicious) participants with which they have had transactions. (iii) On the contrary, a good participant and a malicious participant will most likely give very different feedback ratings to the same set of participants whom they have interacted with. Thus, in GroupTrust, we adopt the concept of feedback rating credibility [4] as a trust behavior indicator to differentiate good participants from dishonest participants.

In the remaining of this subsection, we first describe how we compute the pairwise rating similarity, then we introduce the similarity based rating credibility measure. We compute local trust value of participant i by combining the weighted rating scores participant i gives to other participants j (j=1, ..., i-1, i+1, ..., n) with the rating credibility (rc_{ij}) from i to j as the weight to each rating score r_{ij}.

Pairwise Feedback Similarity. In a network of n participants, each participant i has a rating vector of size n, denoted by $<r_{i1}, r_{i2}, ..., r_{in}>$. To compute the similarity between two rating vectors of participants i and j, we use Weighted Euclidean Distance (WED) based method, which captures the degree of variation or "dispersion" in the historical feedback ratings given by the participants i and j. The larger the dispersion is, the smaller their similarity will be. Thus the feedback based similarity between two participants i and j can be defined as follows:

$$sim(i,j) = \begin{cases} 1 - \sqrt{\sum_{q \in cm(i,j)} w_{(i,j,q)} \cdot (r(i,q) - r(j,q))^2} & if |cm(i,j)| \neq 0 \\ 0 \quad otherwise \end{cases} \tag{5}$$

where $cm(i,j)$ denotes the subset of the common participants that have had interactions with both i and j, $r(i,q)$ denotes the aggregate rating score that i gives to q, and $w_{(i,j,q)}$ denotes the contribution weight of participant q on the similarity measure $sim(i,j)$ defined by the normalized standard deviation of $r(i,q)$ and $r(j,q)$ over their mean, defined by $Avg(i,j,q)$, as follows:

$$\begin{aligned} e_{(i,j,q)} &= \sqrt{\frac{(r(i,q) - Avg(i,j,q))^2 + (r(j,q) - Avg(i,j,q))^2}{2}} \\ Avg(i,j,q) &= \frac{r(i,q) + r(j,q)}{2} \\ w_{(i,j,q)} &= \begin{cases} \frac{e_{(i,j,q)}}{\sum_m e_{(i,j,m)}} if \sum_m e_{(i,j,m)} \neq 0 \\ 0 \quad otherwise \end{cases} \end{aligned} \tag{6}$$

For feedback similarity computation, we use the WED instead of traditional Euclidean Distance (ED) ($w_{(i,j,1)} = \cdots = w_{(i,j,|cm(i,j)|)} = 1/|cm(i,j)|$) [4], [12]. This is because some strategically malicious participants may be dishonest in feedback ratings while providing good services at f% of time (camouflage). These types of malicious participants behave the same as good participants in performing transactions (services), however, unlike good participants, these strategically malicious participants expose themselves in providing dishonest ratings: give bad ratings to good participants and good ratings to the collusive chain and collective of bad participants, as described in Threat Models C and D. We use the rating similarity as an effective countermeasure to define the pairwise rating credibility and use this rating credibility to amplify the pairwise dissimilarity.

We use an example to illustrate the rationality of WED based similarity. Assume the feedback rating vectors of participants i and j are $<$0.10, 0.30, 0.02, 0.05$>$ and $<$0.01, 0.05, 0.05, 0.85$>$. The traditional ED based similarity is 0.578, while WED based similarity is 0.328. Based on the analysis on the two vectors, we observe that i and j should be dissimilar. This is captured by our rating similarity measure.

Computing Similarity based Feedback Rating Credibility. Using the pairwise rating credibility to weight the local trust computation in GroupTrust, we can constrain the malicious participants who have received high feedback ratings not to misuse their high rating scores by using the pairwise rating credibility as the weight to the local trust computation. We below utilize the exponent to define the pairwise similarity based rating credibility:

$$rc_{ij} = e^{(1-1/sim(i,j))} \tag{7}$$

This formula indicates that the feedback rating credibility is exponentially constrained such that the rating credibility is high when the pairwise similarity is high, and vice versa.

Let cf_{ij} denote the rating credibility weighted local trust value that participant i has placed on participant j. We can compute cf_{ij} as follows:

$$cf_{ij} = rc_{ij} \cdot r_{ij} = e^{(1-1/sim(i,j))} \cdot r_{ij} \tag{8}$$

This rating credibility weighted local trust computation ensures that a participant has high local trust value only if this participant has received both high rating credibility (rc_{ij}) and high transaction ratings (r_{ij}) simultaneously.

Given that good participant i and malicious participant j are largely dissimilar in their feedback rating behavior, even if the bad participants have high feedback ratings r_{ij} by strategically behaving with camouflage or spying, their rating credibility (rc_{ij}) is relatively small. By Formula (8), the weighted local trust cf_{ij} will be much smaller than r_{ij}.

In summary, by using similarity based rating credibility as a weight to the aggregate transaction rating score, the GroupTrust model can effectively reduce the role of positive ratings that malicious participants have received from good participants by means of camouflage or malicious spies. This countermeasure is effective because using the pairwise similarity based rating credibility as a weight to the aggregate rating score, the local trust computation can effectively capture the sharp difference between good participants and malicious participants in both transaction based ratings they give to other participants and the feedback ratings that they have received from others. Thus, our GroupTrust model is significantly more effective and more dependable than EigenTrust [5] and many other existing trust models under the Threat Models C and D while performing equally well under simple attacks such as Threat Models A and B.

3.2 Controlled Trust Propagation

Although we can restrain malicious participants from gaining high local trust by using feedback rating credibility, we cannot completely block the trust propagation from good participants to malicious participants using the uniform trust propagation kernel. Therefore, the second novelty of our GroupTrust is to introduce a controlled trust propagation scheme. The main idea is based on the wide consensus that in most collaborative systems participants often do not have the same response to the same stimuli or trust scores.

To reflect such dynamics, an ideal trust model should support differential trust propagation for different participants and for the same participant at different iteration rounds. In GroupTrust, we adapt the *Susceptible-Infected-Recovered* (SIR) propagation model to control when and how much a participant propagates trust to another participant.

Susceptible-Infected-Recovered based Trust Propagation. The SIR model can be seen as an epidemic prediction model, which has been studied and applied in several scientific fields [13], [14], [15], [16]. In an SIR model, each participant in an interactional community can be either susceptible (S) or infected (I): a susceptible participant can become infected when in contact with another infected participant, and it can also heal itself with a certain probability of being recovered (R) and becomes susceptible once again. Thus, one can learn how infection spreads along the direct edges of a network over time.

In GroupTrust, we model the controlled trust propagation process as a dynamic SIR process in which we model the dynamic propagation states of participants by following the Susceptible-Infected-Recovered based propagation model, aiming at breaking the static assumption of uniform trust propagation in which participants with the same trust score will propagate their trust to their neighbors regardless of the states of their neighbors, e.g., whether they are good or bad and have similar or dissimilar transaction/interaction behavior or rating behavior. By utilizing the SIR model, we can capture the dynamics of interaction/transaction/rating behavior of different participants by two states, S and I, with recovered probability to control the state transition. For example, a participant can exhibit different propagation behavior at different rounds of trust propagation iterations by using the SIR model. Different participants with the same trust score may have different propagation decisions based on their own transition state and their neighbors' transition state for trust propagation.

Concretely, we define a discrete contact-based process with two states, S and I. We use susceptible (S) state to refer to the scenario where (1) participants with state S can receive trust propagation from its in-edge neighbors that are in state I, though they cannot propagate their trust scores to their neighbors; and (2) participants in state I can propagate their trust scores to their out-edge neighbors that have state S, though they cannot receive trust from its in-edge neighbors until their state becomes S again. Recall the SIR model in [13], the participants with state I are infected and thus can cause its neighbor participants with state S to be infected but do nothing to its neighbors that are already infected (in state I). Also at each time step, an infected participant makes a number of trials to transmit the infection to its neighbors with probability β per unit time, and at same time this infected participant can heal itself and be recovered at rate μ and get back to the susceptible state. The probability that a participant contacts another participant represents the strength to transmit infection. It is differential to different pairs of connected participants. Using the same analogy, in GroupTrust, we set contact probability from participant i to participant j by the normalized aggregate rating r_{ij}. Theoretically, a threshold exists in the SIR model [13], which determines whether an infection will eventually become either prevalent or extinct. GroupTrust aims to figure out such a threshold to control trust propagation, therefore we propose an SIR-based threshold control mechanism to determine when and how much a participant propagates trust to another participant. We use state I to refer to those participants that are able to propagate trust to their out-edge neighbors and participants with state I may have probability to stop their trust propagation by changing their state from I to S. This dynamic state transition during the iterative trust propagation process allows us to learn how the trust (infection) spreads along the direct edges of the rating network over time. Thus, we can compute the trust score of a participant at the (k+1)th round of iterations in the trust propagation process by the discrete-time probability that a participant is infected (i.e., receiving trust propagation) at the (k+1)th round of iterations:

$$\begin{aligned} t^{(k+1)}(i) = (1 - h^{(k)}(i)) \cdot (1 - t^{(k)}(i)) + (1 - \mu) \cdot t^{(k)}(i) \\ + \mu \cdot (1 - h^{(k)}(i)) \cdot t^{(k)}(i) \end{aligned} \tag{9}$$

where $h^{(k)}(i)$ denotes the probability that participant i is not be infected by any of its neighboring participant(s).

$$h^{(k)}(i) = \prod_{j=1}^{N} (1 - \beta \cdot r_{ji} \cdot t^{(k)}(j)) \tag{10}$$

Formula (9) shows three considerations: (i) participant i is susceptible (1-$t^{(k)}(i)$) and infected (1-$h^{(k)}(i)$) by at least one neighboring participant; (ii) participant i is in infected state and has not recovered; (iii) participant i is in infected state and recovers $\mu \cdot t^{(k)}(i)$, but re-infected by at least one neighbor participant (1-$h^{(k)}(i)$). It assumes the most general case where recovery and infection occur concurrently.

Threshold of Trust Propagation. As mentioned above, given the transmit probability β and the recovered rate μ, a critical threshold τ exists. In the SIR model, this threshold τ determines whether an infection will eventually become either prevalent or extinct. In GroupTrust this τ controls the tendency of trust diffusion such either: (i) the trust dies out; or (ii) the trust becomes prevalent over time. From Formula (9), we can see two critical constants play an important role in propagating trust: transmit probability β and recovered rate μ. The former encourages the propagation from one participant to another participant; and the latter inversely tries to block this propagation. Next we deeply conduct a formal study on how these two variables affect the trust propagation in theory. We first define the threshold τ.

Definition. The threshold τ is such a value that: i) if $\frac{\beta}{\mu} < \tau$, then the trust in the entire network gradually dies out over time, this implies that the trust scores for all the participants will tend to be zero; ii) if $\frac{\beta}{\mu} > \tau$, then the trust gradually becomes prevalent, this means the trust will tend to be stable as the number of iteration rounds increases.

Upon this definition, we prove the following theorems.

Theorem 1. In GroupTrust, the critical threshold τ is:

$$\tau = \frac{1}{\lambda_{1,R}} = 1 \tag{11}$$

where $\lambda_{1,R}$ is the maximum eigenvalue of contact probability matrix R.

We below provide theorems and proofs on the necessity and sufficiency of this threshold. This formal analysis is inspired by [16] with some revision since the network in

[16] is undirected and the rating network in GroupTrust is directed, some properties that are hold for undirected graph may not be adequate for the directed graph.

Theorem 2 (Necessity of Threshold). To guarantee the trust of each participant tends to be zero, the transmit probability must be less than recovered rate, i.e., $\beta<\mu$.

Proof. We have defined the SIR-based Trust as:

$$\begin{aligned} t^{(k+1)}(i) &= (1-h^{(k)}(i))\cdot(1-t^{(k)}(i)) + (1-\mu)\cdot t^{(k)}(i) \\ &\quad + \mu\cdot(1-h^{(k)}(i))\cdot t^{(k)}(i) \\ &= 1-(1-t^{(k)}(i)) \ \cdot h^{(k)}(i) - \mu\cdot t^{(k)}(i)\cdot h^{(k)}(i) \end{aligned} \tag{12}$$

Alternatively, we can formulate Formula (12) using a matrix form:

$$\overrightarrow{T}^{(k+1)} = F(\overrightarrow{T}^{(k)}) \tag{13}$$

Thus, we have:

$$\begin{aligned} F_i(\overrightarrow{T}^{(k)}) &= 1-(1-t^{(k)}(i)) \ \cdot h^{(k)}(i) - \mu\cdot t^{(k)}(i)\cdot h^{(k)}(i) \\ &= 1-\prod_j^N (1-\beta\cdot r_{ji}\cdot t^{(k)}(j)) \\ &\quad + t^{(k)}(i)\cdot\prod_j^N (1-\beta\cdot r_{ji}\cdot t^{(k)}(j)) \\ &\quad - \mu\cdot t^{(k)}(i)\cdot\prod_j^N (1-\beta\cdot r_{ji}\cdot t^{(k)}(j)) \end{aligned} \tag{14}$$

If $\overrightarrow{T}^{(k)}=\overrightarrow{0}$ for all the participants, the $\overrightarrow{T}^{(k+1)}$ will be $\overrightarrow{0}$, and trust will die out, thus $\overrightarrow{T}^{(k)}=\overrightarrow{0}$ is a fixed point of system. Nevertheless, the question is whether the fixed point $\overrightarrow{T}^{(k)}=\overrightarrow{0}$ is asymptotically stable. If stable, the trust will decline to zero and finally die out; if not, then the trust will become prevalent. Utilizing the following lemma, we can capture the critical threshold.

Lemma (Asymptotic Stability). The entire network is asymptotically stable at $\overrightarrow{T}^{(k)}=\overrightarrow{0}$ if the eigenvalues of $\nabla F(\overrightarrow{0})$, where $[\nabla F(\overrightarrow{0})]_{ij} = \frac{\partial F_i}{\partial t_j}|_{\overrightarrow{T}=\overrightarrow{0}}$, are less than 1 in the absolute value.

From formula (14), we have:

$$[\nabla F(\overrightarrow{0})]_{ij} = \begin{cases} \beta r_{ji} & for j\neq i \\ 1-\mu & for j=i \end{cases} \tag{15}$$

equivalently, $\nabla F(\overrightarrow{0}) = \beta R^T + (1-\mu)I$ (I: identity matrix).

Next, we call $\nabla F(\overrightarrow{0})$ as system matrix and define it as:

$$S = \nabla F(\overrightarrow{0}) = \beta R^T + (1-\mu)I \tag{16}$$

Based on the proof of Lemma 2 in [16], we can know that contact probability matrix R and system matrix S have same eigenvectors, and eigenvalues are related:

$$\lambda_{i,S} = 1-\mu+\beta\lambda_{i,R} \quad \forall i \tag{17}$$

Hence, referring to the stability constraint above, the system is asymptotically stable when

$$|\lambda_{i,S}| < 1 \quad \forall i \tag{18}$$

For the further proof, we present Perron-Frobenius theorem [17], [18] as follows: a nonnegative matrix $M\in R^{n\times n}$ has an eigenvalue λ_p that is real and nonnegative with associated nonnegative left and right eigenvectors, and all other eigenvalues are smaller than or equal to λ_p in modulus.

According to this Perron-Frobenius theorem, as the contact probability matrix R is a nonnegative, its maximum eigenvalue $\lambda_{1,R}$ will be nonnegative and greater than or equal other eigenvalues in modulus:

$$\lambda_{1,R} = |\lambda_{1,R}| \geq |\lambda_{i,R}| \quad \forall i \tag{19}$$

Since $0\leq$ 1- $\mu\leq$1 and $0\leq\beta\leq$1, the maximum eigenvalue of system matrix S:

$$\lambda_{1,S} = |\lambda_{1,S}| = |1-\mu+\beta\lambda_{1,R}| \geq |\lambda_{i,S}| \quad \forall i \tag{20}$$

Keeping the stability constraint in formula (18), we make the maximum eigenvalue $\lambda_{1,S}$ less than 1:

$$1-\mu+\beta\lambda_{i,R} < 1 \tag{21}$$

Thus, the trust will become stable eventually and dies out over time when $\frac{\beta}{\mu}<\frac{1}{\lambda_{1,R}}$, and the threshold is $\tau=\frac{1}{\lambda_{1,R}}$.

Let M $(m_{ij}\geq 0)$ be $n\times n$ nonnegative matrix without zero row, $\phi_i(M)=\sum_{k=1}^{n} m_{ik}$ and $\varphi_j(M)=\sum_{k=1}^{n} m_{kj}$ denote the sums of ith row and jth column. Then the maximum eigenvalue γ of matrix M is subject to the constraint [18]:

$$\min_i \phi_i(M) \leq \gamma \leq \max_i \phi_i(M) \tag{22}$$

In GroupTrust, the contact probability matrix R is a stochastic matrix, namely the sum of each row is 1, thus $1\leq\gamma\leq$1, this indicates the maximum eigenvalue γ of contact probability matrix R is 1. Thus we prove that the critical threshold is $\tau=\frac{1}{\lambda_{1,R}}$=1. This implies the trust will become prevalent when $\beta>\mu$, or dies out when $\beta<\mu$.

Theorem 3 (Sufficiency of Threshold). If $\frac{\beta}{\mu}<\tau$=1, then the trust will tend to be zero gradually, irrespective the initial trust score of each participant.

Proof. Since parameters β, r_{ij} and $t^{(k)}(i)$ are nonnegative and less than 1, we have:

$$\begin{aligned} h^{(k)}(i) &= \prod_{j=1}^{N}(1-\beta\cdot r_{ji}\cdot t^{(k)}(j)) \\ &\geq 1-\beta\cdot\sum_{j=1}^{N} r_{ji} \ \cdot t^{(k)}(j) \end{aligned} \tag{23}$$

For each participant i=1, ..., n,

$$\begin{aligned} t^{(k+1)}(i) &= \ 1-(1-t^{(k)}(i)) \ \cdot h^{(k)}(i) - \mu\cdot t^{(k)}(i)\cdot h^{(k)}(i) \\ 1-t^{(k+1)}(i) &= \ (1-t^{(k)}(i)) \ \cdot h^{(k)}(i) + \mu\cdot t^{(k)}(i)\cdot h^{(k)}(i) \\ &= (1-(1-\mu)t^{(k)}(i))\cdot h^{(k)}(i) \\ &\geq (1-(1-\mu)\cdot t^{(k)}(i))\times(1-\beta\cdot\sum_{j=1}^{N} r_{ji}\cdot t^{(k)}(j)) \\ &\geq 1-(1-\mu)\cdot t^{(k)}(i) - \beta\cdot\sum_{j=1}^{N} r_{ji}\cdot t^{(k)}(j) \end{aligned} \tag{24}$$

$$t^{(k+1)}(i) \leq (1-\mu)\cdot t^{(k)}(i) + \beta\cdot\sum_{j=1}^{N} r_{ji}\cdot t^{(k)}(j) \tag{25}$$

By utilizing the system matrix S, we can rewrite the formula (25) in a vector form:

$$\overrightarrow{T}^{(k+1)} \leq S\overrightarrow{T}^{(k)} \leq S^2\overrightarrow{T}^{(t-1)} \leq \cdots \leq S^{(k+1)}\overrightarrow{T}^{(0)} \tag{26}$$

Because the initial trust is irrespective, we assume the initial trust of system participants as:

$$\overrightarrow{T}^{(0)} = \psi_1\overrightarrow{x_1} + \psi_2\overrightarrow{x_2} + \cdots + \psi_n\overrightarrow{x_n} \tag{27}$$

where $\vec{x_i}$ (i=1,..., n) are the eigenvectors corresponding to the eigenvalues $\lambda_{i,S}$(i=1, ..., n), and ψ_1>0. The eigenvector corresponding to maximum eigenvalue 1 of stochastic matrix is $(1, ..., 1)^T$. As we know, the contact probability matrix R and system matrix S have same eigenvector, thus one of its eigenvectors is (1, ..., 1) as well, this guarantees that the initial trust scores are not zero.

Thus, we have:

$$\begin{aligned}\vec{T}^{(k+1)} &\leq S^{(k+1)}\vec{T}^{(0)} \\ &= \psi_1 \cdot S^{(k+1)} \cdot \vec{x_1} + \psi_2 \cdot S^{(k+1)} \cdot \vec{x_2} \\ &\quad + \cdots + \psi_n \cdot S^{(k+1)} \cdot \vec{x_n} \\ &= \psi_1 \cdot \lambda_{1,S}^{(k+1)} \cdot \vec{x_1} + \psi_2 \cdot \lambda_{2,S}^{(k+1)} \cdot \vec{x_2} \\ &\quad + \cdots + \psi_n \cdot \lambda_{n,S}^{(k+1)} \cdot \vec{x_n} \\ &= \lambda_{1,S}^{(k+1)} \cdot (\psi_1 \cdot \vec{x_1} + (\tfrac{\lambda_{2,S}}{\lambda_{1,S}})^{(k+1)} \cdot \psi_2 \cdot \vec{x_2} \\ &\quad + \cdots + (\tfrac{\lambda_{n,S}}{\lambda_{1,S}})^{(k+1)} \cdot \psi_n \cdot \vec{x_n})\end{aligned} \tag{28}$$

As we know, the maximum eigenvalue $\lambda_{1,S}^{(k+1)}$ is non-negative, namely $(1 - \mu + \beta\lambda_{1,R})|_{\lambda_{1,R}=1}$, and greater than or equal to the absolute value of other eigenvalues. Since

$$\begin{aligned}\lambda_{1,S} &= 1 - \mu + \beta\lambda_{1,R} \\ &< 1 - \mu + \beta \cdot \tfrac{\mu}{\beta} \quad (\lambda_{1,R} < \tfrac{\mu}{\beta}) \\ &< 1\end{aligned} \tag{29}$$

$\lambda_{1,S}^{(k+1)} \approx 0 \Rightarrow S^{(k+1)}\vec{T}^{(0)}=0$, indicating the trust of entire network decays exponentially and declines to zero over time.

Global Trust Computation. By the above theoretical analysis, we can see that trust established through SIR-controlled propagation confronts two outcomes in the end, either extinction or prevalence. In GroupTrust, we aim at figuring out differential propagations for participants with different transaction/interaction/rating behavior, namely, enabling good participants to gain trust propagation with high probability, blocking malicious participants to gain trust propagation with low or zero probability.

In order to achieve this goal, we define a fined-grained trust propagation threshold for each pair of connected participants instead of the coarse-grained constant threshold. We replace the system-wide constant transmit probability and recovered rate in [13], [14], [15], [16] by two variables to reflect the fact that different pairs of connected participants may have different transmit probability and recovered rate. Hence, we redefine the global trust computation formula as:

$$\begin{aligned}t^{(k+1)}\ (i) &= 1 - \prod_j^N (1 - \beta_{ji} \cdot r_{ji} \cdot t^{(k)}(j)) \\ &+ t^{(k)}(i) \cdot \prod_j^N (1 - \beta_{ji} \cdot r_{ji} \cdot t^{(k)}(j)) \\ &- \mu_{ij} \cdot t^{(k)}(i) \cdot \prod_j^N (1 - \beta_{ji} \cdot r_{ji} \cdot t^{(k)}(j))\end{aligned} \tag{30}$$

In this formula, for each pair of connected participants, we introduce pairwise transmit probability β_{ij} and recovered rate μ_{ij}. Based on Theorems 2 and 3, if transmit probability is bigger than recovered rate, the trust propagation is permitted, otherwise blocked. We also observe that the good participants are similar with other good participants, but dissimilar with malicious participants. Therefore, we define the pairwise transmit probability β_{ij} using the similarity based rating credibility weighted local trust cf_{ij}, which impels good participants to propagate trust to good participants, constraining malicious participants from gaining trust propagation by strategically malicious collusion. Given that some strategically malicious participants may have a positive similarity with good participants, such as camouflage participants in Threat Model C or spy participants in Threat Model D, we utilize recovered rate to define a pairwise propagation threshold to block malicious trust propagation. In order to coordinate with the scope of transmit probability, we define recovered rate in the range of [0, 1] by *max-min* method:

$$\mu_{ij} = \frac{1/(1+e^{sim(i,j)}) - 1/(1+e^{max(sim(u,v))})}{1/(1+e^{min(sim(u,v))}) - 1/(1+e^{max(sim(u,v))})} \tag{31}$$

where $max(sim(u, v))$ is 1.0 and $min(sim(u, v))$ is 0.0. This formula shows that the smaller the pairwise rating similarity is, the higher the propagation threshold will be and vice versa. The rating similarity between two participants i and j is computed by how consistent their feedback ratings to the common set of participants with which both have had transactions in the past independently. The higher similarity implies more consistency in their feedback ratings over the common set of other participants. Thus, a smaller pairwise similarity will lead to a higher propagation threshold μ_{ij} and vice versa. By using the similarity metric, we block trust propagation between dissimilar participants to a large extent, and promote trust propagation between participants with similar trust and rating behavior.

In summary, the global trust computation consists of two phases: the local trust computation and the controlled trust propagation. GroupTrust is unique in two aspects. First, its local trust computation enhances the existing rating aggregation methods by incorporating pairwise feedback rating credibility as the weight to the normalized aggregate rating score. Second, its trust propagation enhances the existing uniform trust propagation models, represented by EigenTrust. By defining the propagation control threshold, it allows GroupTrust to capture the dynamic trust propagation behavior: (i) different participants with even the same local trust value may have different propagation threshold, depending whether connected participants share similar transaction/rating behavior; and (ii) each participant may have different thresholds at different rounds of iteration during trust propagation, depending on both its own propagation threshold and the thresholds of its neighbors.

4 Experimental Evaluation

In this section, we conduct extensive experiments with synthetic and real world datasets to evaluate the efficiency and effectiveness of GroupTrust by comparing with Non-Trust and four representative trust models, EigenTrust [5], a recently developed trust propagation model, ServiceTrust [12], feedback rating credibility weighted local trust with uniform trust propagation, denoted by GroupTrust_UP, and the SIR controlled trust propagation with the simple aggregate rating based local trust (see Formula (1)), denoted by GroupTrust_RLT. We use GroupTrust to refer to the trust model powered by both rating credibility weighted local trust and the SIR controlled trust propagation.

TABLE 1
Experimental Configuration

Network Structure	number of total participants in Threat Models A, B, C and D	600, 600, 700, 1000
	number of pre-trusted participants	30
	number of initial neighbors of good, malicious and pre-trusted participants	2, 10 and 10
	number of hops for query process	7
File Distribution	file distribution at good participants	Zipf distribution over 200 distinct files
	number of distinct files at good participant	uniform random distribution
	top % queries for most popular files pre-trusted participants respond to	5%
	% file categories owned by good participants in Threat Model A, B and C	15%
	% file categories owned by good participants in Threat Model D	10%
	% file categories owned by malicious participants in Threat Model A, B, D	100%
	% file categories owned by malicious participants in Threat Model C	55%
Participant Behavior	% download requests in which good participant returns inauthentic file	5%
	downloads source selection method	probabilistic algorithm
	probability that participants with global trust score 0 are selected	range [0-10%]

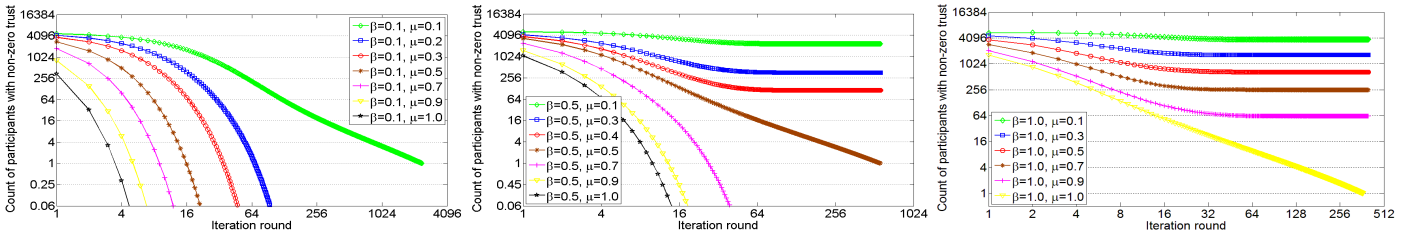

Fig. 3. Trust propagation threshold with different sets of transmit probabilities and recovered rates (log-log scales).

The TM/RM [19] simulator is an open-source program. The main component of this TM/RM simulator is to simulate the peer-to-peer (P2P) transactional environment for query answering services among participants such that the need to create a working P2P file sharing system is avoided. However, TM/RM does not provide the strategically malicious attack models such as Threat Model C and Threat Model D. Also it does not support for pairwise rating similarity computation. Thus, we develop our simulator to simulate Threat Models C and D, perform pairwise similarity computation, and conduct SIR-related experiments using the Epinions real dataset to compare different trust propagation models. We adopt the same setting as that in EigenTrust with respect to the total number of participants used in the experiments for all four threat models, as shown in Table 1 for the small scale experiments. We also extend the small setup by using 10 times of the participants in the small scale setting for good, bad and pre-trusted participants, which results in 630 total number of participants in Threat Models A and B, 200 disguise malicious participants, 500 good participants and 30 pre-trusted participants in Threat Model C. For Threat Model D, 400 type B and type D malicious participants, 600 good participants and 30 pre-trusted participants.

4.1 Trust Propagation Threshold

In this section we verify the effect of the trust propagation threshold using Epinions, a real world dataset. We conduct our experiments using the Epinions network of 10,000 nodes in which the contact probability between each pair of connected participants is generated by Zipf distribution, and the initial global trust for each participant is set to 0.5. With this configuration, we perform GroupTrust with SIR based trust propagation. Fig. 3 depicts the experimental results in terms of different settings of transmit probabilities (0.1, 0.5 and 1.0) and recovered rates (0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 0.9 and 1.0). We make four observations from this set of experiments. (1) There is a clear separation for each setting of β and μ, when $\beta<\mu$, the number of participants with non-zero global trust score goes to zero as the iteration round increases. (2) As μ increases, it takes more number of iteration rounds for the trust score approaching zero. (3) When $\beta>\mu$, the number of participants with non-zero global trust tends to be a certain ratio, showing the property that the global trust is becoming prevalent. (4) With fixed β, as μ decreases, the difference between β and μ gets bigger, and it takes less number of iteration rounds to reach convergence for a large number of participants receiving non-zero trust. This set of experiments verifies the effect of trust propagation threshold.

We have studied the impact of different pairings of β and μ on the total number of participants with non-zero trust. This study helps us understand how different β and μ pairs may impact on the results of trust computation of the participants. In GroupTrust, we define the transmit probability β_{ij} for each pair of participants i and j, by using the rating credibility weighted local trust cf_{ij} defined in Formula (8), and recovered rate μ_{ij} as the pairwise propagation threshold, defined by Formula (31) in Section 3.2 on SIR controlled trust propagation. The value of β and the value of μ are computed for each pair of participants in all experiments reported in Section 4.2, 4.3, 4.4.

4.2 Performance Comparison

We conduct four groups of experiments to compare GroupTrust with Non-Trust, EigenTrust, ServiceTrust, GroupTrust_UP and GroupTrust_RLT under the four threat models. We set the number of transactions as 10 times the number of system participants, such that the total number of participants in Threat Models A and B are 630 (30 pre-trusted participants), wherein the ratios of malicious participants are set as 0, 10%, 30%, 50% and 70%. For Threat Model C, the disguise, good and pre-trusted participants are 200, 500 and 30. The evaluation measures the resilience of the trust models under varying camouflage percentage f under a constant ratio of malicious participants at 27%. For Threat Model D, the malicious participants (type B and type D with different combinations), good and pre-trusted participants are 400, 600 and 30. The evaluation of trust models is performed by varying the ratio of type D and type B malicious participants under a constant ratio of malicious participants at 39%. Fig. 4 reports our experimental results. We make two observations. First, we observe that all five trust models offer the same level of effectiveness under Threat Models A and B. Given that pre-trusted participants have non-zero initial trust scores $1/|P|$ in EigenTrust, under Threat Models A and B, the malicious participants can never gain positive ratings from other participants because bad participants have very dissimilar behavior in both transaction and rating, thus their global trust scores are always zero no matter whether Formula (2) or Formula (30) are used.

Second, we observe that GroupTrust, GroupTrust_UP, GroupTrust_RLT and ServiceTrust significantly outperform EigenTrust under Threat Models C and D, when bad participants strategically cheat by providing good transaction services at f% cases or by selecting certain bad participants as spy participants in both services and ratings. Concretely, by comparing ServiceTrust and GroupTrust_UP with EigenTrust, we show that the use of feedback credibility weighted local trust computation is effective in degradation of the effect of malicious manipulation of feedback ratings via strategically malicious camouflage or spying behavior. Similarly, by comparing GroupTrust and GroupTrust_RLT with EigenTrust, we show the effectiveness of using the SIR-based threshold controlled trust computation in minimizing or removing the effect of malicious manipulation.

To further understand the difference of GroupTrust with GroupTrust_UP, GroupTrust_RLT and ServiceTrust, we zoom in the comparison of these four models in the small embedded figure in Fig. 4(c) to compare the controlled trust propagation with the uniform trust propagation adopted in GroupTrust_UP and ServiceTrust. We observe that GroupTrust outperforms both GroupTrust_UP and ServiceTrust, showing the effect of controlled trust propagation and the role that the controlled propagation plays in trust establishment, maintenance and management. We also observe that GroupTrust_RLT has the same level of effectiveness as GroupTrust, showing that SIR controlled propagation model combined with the rating similarity based propagation threshold is by itself effective in countering the four attack models, though GroupTrust provides more resilient local trust computation and thus higher quality for trust propagation and faster convergence rate.

4.3 Scalability Evaluation

This subsection evaluates the effectiveness of GroupTrust in terms of its scalability with respect to different network sizes. We have analyzed that one of the common weaknesses of existing trust models [1], [2], [4], [5] is the poor performance when the rating network is large and sparse.

The first set of experiments measures the effectiveness of GroupTrust by comparing it with Non-Trust and four different trust models, EigenTrust, ServiceTrust GroupTrust_UP, GroupTrust_RLT with varying the network sizes, Fig. 5(a) shows the results under Threat Model C in which the percentage of malicious participants is 27%, the camouflage probability f is 40%, and the number of transactions in each network is set to be 10 times the number of participants. Fig. 5(a) shows that as the network size increases, GroupTrust scales well and the fraction of inauthentic downloads remains consistently low. In contrast, EigenTrust not only has much higher amount of inauthentic downloads for networks of different sizes but also the amount of inauthentic downloads is increased gradually as the size of the rating network increases and when the size of the network is at 1,000 nodes or higher, the fraction of inauthentic downloads in EigenTrust is close to that of Non-Trust. This implies that the trust model by employing normalized aggregate ratings to compute local trust combined with uniform propagation is inadequate for computing trust in the presence of strategically malicious participants. Due to the space constraint, we omit the evaluation results for Threat Model D, as similar results can be observed.

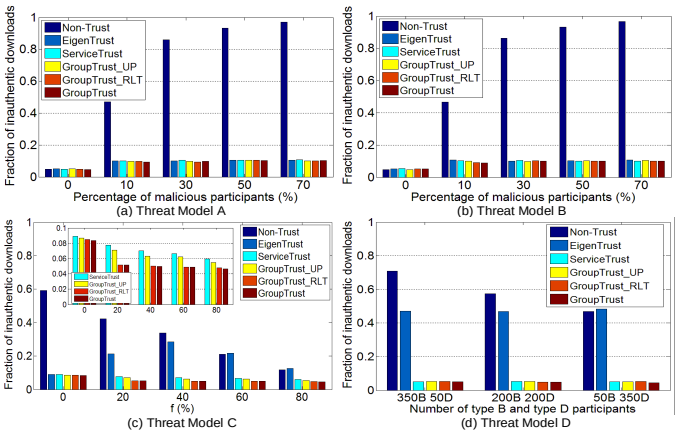

Fig. 4. Performance evaluation under Threat Models A, B, C and D.

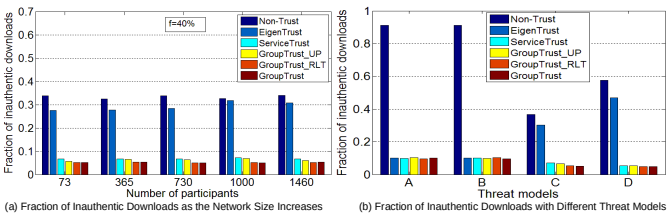

Fig. 5. Evaluation on scalable and equal networks.

The second set of experiments measures the performance of the four attack models under the same network setting with the total number of participants fixed at 600 good participants, 400 malicious participants and 30 pre-trusted participants. We set f = 40% in Threat Model C, and make type B and type D equal in Threat Model D. Fig. 5(b) shows the experimental results. As analyzed in Section 2.4, the global trust scores of malicious participants are zero in Threat Models A and B, and the fraction of inauthentic downloads keeps no significant change with size of the network. For Threat Model C, even as the ratio of disguise participants becomes larger, there is no impact on GroupTrust_RLT and GroupTrust, thanks to the controlled trust propagation. However, the fraction of inauthentic downloads increases as the ratio of malicious camouflage participants rises for all uniform propagation based trust models, such as EigenTrust, ServiceTrust and GroupTrust_UP. This shows that the strategically malicious participants can do more harm compared to those isolated and non-disguise malicious participants.

4.4 Evaluation with Epinions Dataset

In this subsection, we evaluate the performance of GroupTrust using the real world datasets Epinions [7]. We focus our evaluation from the perspective of attack resilience under Threat Models C and D and computation complexity. We set the number of pre-trusted participants to 3% of system nodes for all experiments reported in this section. Given that pre-trusted participants serve as the central authority of the system, we choose the 3% largest in/out-degree nodes as the pre-trusted participants.

Attack Resilience. Given that the malicious participants in Threat Models C and D can gain positive feedback ratings through providing good services, we add some strategically collusive participants to the Epinions dataset to evaluate the effectiveness of GroupTrust. Concretely, we add 10, 30 and 50 malicious participants into the Epinions network, organized by 100 regular nodes, and make these camouflage participants form a chain with 1.0 feedback ratings over each link as defined in Threat Model C. For Threat Model D, we add 30 malicious participants organized in three cases: 10 type B and 20 type D, 15 type B and 15 type D, and 20 type B and 10 type D. In order to make these malicious participants receive sufficient amount of feedback ratings, we connect them to the top 10 highest degree participants in the network. We set a range [0, 0.05] from which malicious participants select feedback ratings, and a range [0.85, 1] from which good participants select feedback ratings.

Fig. 6 shows the experimental results. The notation "100G+30M" denotes the network organized by 100 good participants and 30 malicious participants, "100G+10B+20D" denotes the network with 100 good participants, 10 type B participants and 20 type D participants. We make three interesting observations. First, Fig. 6 shows that GroupTrust and GroupTrust_RLT significantly outperform EigenTrust, ServiceTrust and GroupTrust_UP, for varying camouflage percentage f under Threat Model C in Fig. 6(a), and for varying ratios of spy (type D) participants and type B malicious participants in Fig. 6(b). This is primarily due to the fact that GroupTrust and GroupTrust_RLT use

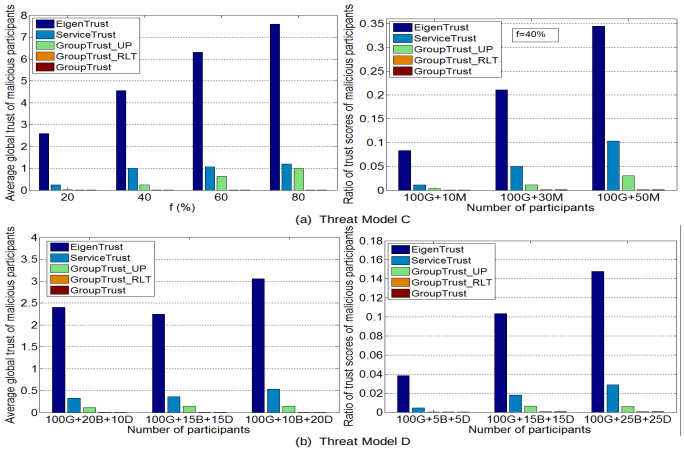

Fig. 6. Trust score and influence evaluation on malicious participants.

SIR-based controlled trust propagation kernel, whereas the other trust models are using the uniform propagation kernel. Second, this set of experiments shows that GroupTrust outperforms all other four models, with GroupTrust_RLT as the second best performer, while GroupTrust_UP and ServiceTrust have better performance than EigenTrust. This is primarily due to the incorporation of feedback rating credibility as weight in the local trust computation. Third, the ratio of trust scores of malicious participants over the trust scores of all participants goes up as the number of malicious participants increases. This indicates the influence of malicious participants on the whole network grows as the number of colluding malicious participants increases.

Computation Complexity. In EigenTrust, the time complexity mainly depends on the computation of global trust scores for all participants. Given that for each participant, its trust score is produced by aggregating the ratings it may have from other n-1 participants, thus for n participants, the complexity of computing trust is $O(n^2)$. In GroupTrust_UP and GroupTrust, for a participant, the SIR-controlled propagation needs $O(1)$ time to check whether a connected participant can meet the trust propagation threshold. The check will continue until all its connected participants are checked, and the number of connected participants is no more than n. Thus, for all the participants in the entire network, the complexity is $O(n^2)$ as well. However, given that some feedback rating credibility weighted local trust can be zero when the pairwise similarities are zero, this implies that some rating relationship links will not be used during the trust propagation, therefore the time complexity in ServiceTrust, GroupTrust_UP and GroupTrust is less than EigenTrust naturally. Moreover, GroupTrust uses a SIR-controlled propagation kernel, for any pair of connected participants, if their pairwise transmit probability is less than their pairwise recovered rate, then the trust propagation will be blocked as well. This further reduces the total number of rating relationship edges used in the global trust computation of GroupTrust. Thus, the time complexity of GroupTrust is lower than that of ServiceTrust and GroupTrust_UP.

Fig. 7 shows the time complexity comparison using Epinions dataset for GroupTrust, EigenTrust, ServiceTrust and GroupTrust_UP by varying network sizes and varying the number of iteration rounds in terms of hops during the

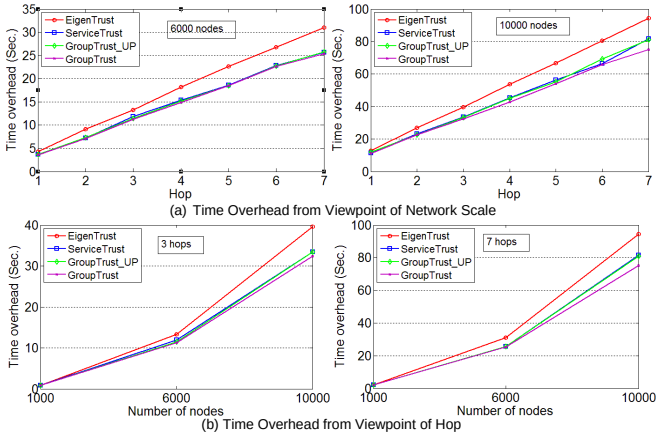

Fig. 7. Time overhead with varying network sizes and hop counts.

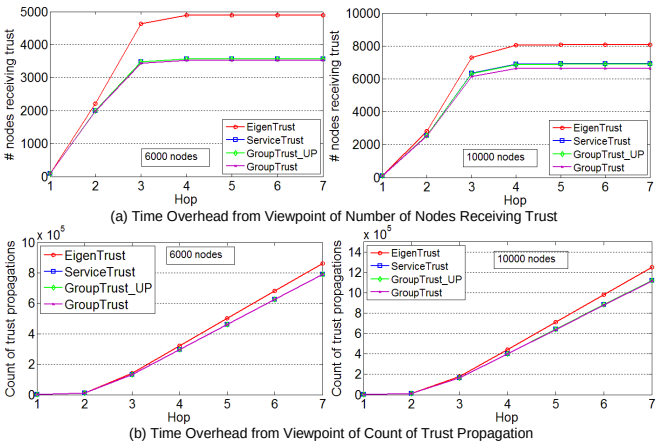

Fig. 8. Time overhead evaluation from viewpoints of number of nodes receiving trust and count of trust propagation.

trust propagation. Fig. 7(a) shows that the time complexity increases as the number of nodes increases from 6,000 to 10,000. Fig. 7(b) shows that the time complexity grows as the propagation scope increases from 3 hops to 7 hops. In general, EigenTrust incurs higher trust propagation overhead than GroupTrust, ServiceTrust and GroupTrust_UP, because EigenTrust propagates trust to all connected neighbors for each participant. In contrast, ServiceTrust, GroupTrust_UP and GroupTrust discard certain pairs of connected participants through two types of filtering during trust propagation: (1) we use pairwise feedback rating credibility as weight to the normalized aggregate rating scores when computing local trust, which filters out those rating relationships that have zero local trust due to their zero similarity scores; and (2) we use SIR-controlled trust propagation, which filters out those pairs of participants in the rating network, when their pairwise transmit probabilities are less than their pairwise recovered rates. Thus the time complexities of GroupTrust, ServiceTrust and GroupTrust_UP are lower than that of EigenTrust, and the time complexity of GroupTrust is lower than that of ServiceTrust and GroupTrust_UP, because GroupTrust supports both filters, but ServiceTrust and GroupTrust_UP support the first filter.

To verify the above statement, we randomly select 5 participants to compute the trust propagation scale, namely, from each of the chosen participants, how many other participants to which its trust can be propagated within 7 hops, and the amount of performing trust propagations under two sizes of Eopinions networks of 6,000 and 10,000 participants. The experimental results are the average value of 5 participants (see Fig. 8). We can see that as the network size grows, GroupTrust takes less time than EigenTrust, ServiceTrust and GroupTrust_UP because it cuts off the trust propagation to malicious nodes through two stage filters, whereas EigenTrust, ServiceTrust and GroupTrust_UP uniformly propagates trust to all neighbors of each participant, no matter whether the neighbors are good or malicious.

5 Related Work

Trust and reputation have been studied by many. In the simple eBay system [6], there are three discrete feedback ratings: positive (1), negative (-1), and neutral (0). One participant's trust score is computed by only aggregating the feedback ratings given by the small subset of participants with whom it has direct transactions. Such direct experience based trust system is easy to implement and fast to compute trust, but it is vulnerable to malicious manipulations due to the presence of dishonest or bad participants. PeerTrust [4] is a classic peer-to-peer trust model, and utilizes pairwise feedback similarity to define the credibility of a participant and compute the trust score of this participant using its credibility as a weight to its rating aggregate. GroupTrust is inspired by PeerTrust's similarity based weighting scheme for the local trust computation.

Most of the trust propagation methods [3], [12] are inspired by EigenTrust [5], which uses the uniform trust propagation kernel to address sparse ratings in large-scale networks. Though the uniform trust propagation model is more effective than those models that compute trust solely based on direct transactional experiences [2], [4], [8] in the presence of sparse ratings (Section 2), EigenTrust and its family of trust models with uniform propagation are vulnerable when there are strategically malicious participants, such as those in Threat Models C and D. GroupTrust is developed as a dependable trust management model for decentralized computing systems. It has two main advantages over other existing models: (i) it incorporates the pairwise similarity based feedback credibility in trust computation; and (ii) it allows trust propagation among only trusted participants through SIR-based controlled trust propagation, which is resilient against strategically malicious attacks in Threat Models C and D. In comparison, PowerTrust [20] leverages the power-law power nodes and look-ahead random walk strategy to speed up the trust propagation, and utilizes the multiple locality preserving hashing (LPH) to prevent malicious participants from reporting wrong global reputation scores. However, PowerTrust is vulnerable under Threat Model C and Threat Model D because bad participants can gain high trust scores through camouflage or malicious spy nodes due to uniform propagation. Similarly, the RLM Trust model [21] proposes a malicious feedback detection mechanism through hypothesis test technology, which evaluates whether the deviation between the feedback reputation and the predicted reputation is normal enough. If the deviation exceeds a certain threshold, the feedback is identified as

malicious, and the update of reputation and the prediction variance are denied. Even though RLM can flag malicious feedbacks, it is vulnerable to type D spy participants in Threat Model D and camouflage participants in Threat Model C due to the lack of propagation control in RLM.

The role of pre-trusted participants is critical in GroupTrust. When new nodes join the system, they have no trust relationship with existing nodes in the system. The pre-trusted nodes can help new nodes to bootstrap their trust with other existing nodes. However, some trust modes, such as SORT [22], do not support the pre-trusted participants. In these trust models, it is hard for new nodes to choose the trustworthy nodes to interact with and it takes much longer for new nodes to establish trust relationships with others in the system.

6 Conclusion

We have presented the design of GroupTrust, a dependable trust management and trust propagation framework. We showed analytically and experimentally that GroupTrust was significantly more attack resilient than existing representative trust models. We address the problems of dishonest ratings and malicious manipulation of ratings to gain high trust values by defining a rating credibility weighted local trust. We address the problem of uniform trust propagation by introducing the SIR-controlled trust propagation kernel to prevent trust propagation from good nodes to bad nodes. Extensive experiments using synthetic and realistic datasets show that GroupTrust significantly outperforms Non-Trust, EigenTrust, ServiceTrust, GroupTrust_UP, GroupTrust_RLT in terms of time complexity and attack resilience in the presence of dishonest and sparse ratings and strategically malicious colluding participants.

References

[1] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision support systems*, vol. 43, no. 2, pp. 618–644, 2007.

[2] J. Golbeck, "Weaving a web of trust," *Science*, vol. 321, no. 5896, pp. 1640–1641, 2008.

[3] Y. Wang and A. Nakao, "Poisonedwater: An improved approach for accurate reputation ranking in p2p networks," *Future Generation Computer Systems*, vol. 26, no. 8, pp. 1317–1326, 2010.

[4] L. Xiong and L. Liu, "Peertrust: supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[5] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *Proceedings of the 12th international conference on World Wide Web*. ACM, 2003, pp. 640–651.

[6] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[7] M. Richardson, R. Agrawal, and P. Domingos, "Trust management for the semantic web," in *The Semantic Web-ISWC 2003*. Springer, 2003, pp. 351–368.

[8] S. Song, K. Hwang, R. Zhou, and Y. Kwok, "Trusted p2p transactions with fuzzy reputation aggregation," *IEEE Internet Computing*, vol. 9, no. 6, pp. 24–34, 2005.

[9] Y. Wang, V. Cahill, E. Gray, C. Harris, and L. Liao, "Bayesian network based trust management," in *Proceedings of the Third international conference on Autonomic and Trusted Computing, Wuhan, China, Sep. 3-6*, vol. 4158. Springer-Verlag New York Inc, 2006, pp. 246–257.

[10] A. Jøsang and R. Ismail, "The beta reputation system," in *Proceedings of the 15th Bled Electronic Commerce Conference*, vol. 160, 2002.

[11] B. Edmonds, E. Norling, and D. Hales, "Towards the evolution of social structure," *Computational & Mathematical Organization Theory*, vol. 15, no. 2, pp. 78–94, 2009.

[12] Z. Su, L. Liu, M. Li, X. Fan, and Y. Zhou, "Servicetrust: trust management in service provision networks," in *2013 IEEE International Conference on Services Computing*. IEEE, 2013, pp. 272–279.

[13] S. Gómez, A. Arenas, J. Borge-Holthoefer, S. Meloni, and Y. Moreno, "Discrete-time markov chain approach to contact-based disease spreading in complex networks," *EPL (Europhysics Letters)*, vol. 89, no. 3, p. 38009, 2010.

[14] S. Gómez, J. Gómez-Gardenes, Y. Moreno, and A. Arenas, "Nonperturbative heterogeneous mean-field approach to epidemic spreading in complex networks," *Physical Review E*, vol. 84, no. 3, p. 036105, 2011.

[15] M. E. J. Newman, "Spread of epidemic disease on networks," *Physical review E*, vol. 66, no. 1, p. 016128, 2002.

[16] D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, and C. Faloutsos, "Epidemic thresholds in real networks," *ACM Transactions on Information and System Security*, vol. 10, no. 4, p. 1, 2008.

[17] C. R. MacCluer, "The many proofs and applications of perron's theorem," *Siam Review*, vol. 42, no. 3, pp. 487–498, 2000.

[18] H. Minc, "Non-negative matrices," *New York*, 1988.

[19] "Tm/rm simulator: http://rtg.cis.upenn.edu/qtm/p2psim.php3."

[20] R. Zhou and K. Hwang, "Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 4, pp. 460–473, 2007.

[21] X. Wang, L. Liu, and J. Su, "Rlm: A general model for trust representation and aggregation," *IEEE Transactions on Services Computing*, vol. 5, no. 1, pp. 131–143, 2012.

[22] A. Can and B. Bhargava, "Sort: A self-organizing trust model for peer-to-peer systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 1, pp. 14–27, 2013.

Xinxin Fan received the B.S. degree from Zhengzhou University of Light Industry, China, the M.S. degree in software engineering, and the Ph.D degree in computer science both from Dalian University of Technology, in 2006, 2009, and 2014 respectively. He was a visiting PHD student in Georgia Tech between 2011 and 2013. Currently, his research interests includes trust management and network security.

Ling Liu is a Professor in the School of Computer Science at Georgia Institute of Technology. She directs the research programs in Distributed Data Intensive Systems Lab (DiSL). Prof. Liu is an IEEE Fellow and a recipient of IEEE Computer Society Technical Achievement Award in 2012. Currently Prof. Liu is the EIC of IEEE Transactions on Service Computing, and serves on the editorial board of half dozen international journals, including TOIT, TWEB, DPD and JPDC.

Mingchu Li received the B.S. degree in mathematics, Jiangxi Normal University and the M.S. degree in applied science, University of Science and Technology Beijing in 1983 and 1989, respectively. He received his doctorate in Mathematics, University of Toronto in 1997. Currently, he worked for School of Software Technology of Dalian University of Technology as a full Professor. His main research interests include theoretical computer science and cryptography.

Zhiyuan Su received the B.S. degree in software engineering, and Ph.D in computer software and theory, in 2002 and 2014, both from the Dalian University of Technology, China. He was a visiting PHD student in Georgia Tech between 2011 and 2013. Now he is a system architect at State Key Laboratory of High-end Server & Storage Technology, China. His current research interests include big data system architechture, cloud computing and trust system.