Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 3610157, 12 pages
http://dx.doi.org/10.1155/2016/3610157

Research Article

A Novel Iterative and Dynamic Trust Computing Model for Large Scaled P2P Networks

Zhenhua Tan,[1] Xingwei Wang,[1,2] and Xueyi Wang[1]

[1]*Software College, Northeastern University, Shenyang 110819, China*
[2]*College of Information Science and Engineering, Northeastern University, Shenyang 110819, China*

Correspondence should be addressed to Zhenhua Tan; tanzh@mail.neu.edu.cn

Received 30 October 2015; Accepted 11 January 2016

Academic Editor: Seung Yang

Trust management has been emerging as an essential complementary part to security mechanisms of P2P systems, and trustworthiness is one of the most important concepts driving decision making and establishing reliable relationships. Collusion attack is a main challenge to distributed P2P trust model. Large scaled P2P systems have typical features, such as large scaled data with rapid speed, and this paper presented an iterative and dynamic trust computation model named IDTrust (Iterative and Dynamic Trust model) according to these properties. First of all, a three-layered distributed trust communication architecture was presented in IDTrust so as to separate evidence collector and trust decision from P2P service. Then an iterative and dynamic trust computation method was presented to improve efficiency, where only latest evidences were enrolled during one iterative computation. On the basis of these, direct trust model, indirect trust model, and global trust model were presented with both explicit and implicit evidences. We consider multifactors in IDTrust model according to different malicious behaviors, such as similarity, successful transaction rate, and time decay factors. Simulations and analysis proved the rightness and efficiency of IDTrust against attacks with quick respond and sensitiveness during trust decision.

1. Introduction

Large scaled P2P network is developed rapidly in recent years because of its openness, anonymity, and being self-organized. P2P networking traffics always occupy Internet and P2P architecture is an important part of future Internet [1]. Various P2P applications, such as Mobile P2P, P2P lending (e.g., Zopa.com), and P2P e-commerce (e.g., eBay), have been welcomed by people and demonstrate high-capacity, variety, and rapid response [2–4].

Trust management has been emerging as an essential complementary part to security mechanisms of P2P systems, and trustworthiness is one of the most important concepts driving decision making and establishing reliable relationships. A well-defined trust model can provide meaningful decision support and help customers reduce possible risks during an Internet transaction. Like trust and reputation in social networks, trust evaluation in P2P systems is based on transaction histories.

Threat and attack always exist in P2P networks. Some common attacks against trust management remain in P2P network, such as Sybil Attack, Newcomer Attack, Betrayal Attack, Inconsistency Attack, Bad-Mouthing/Ballot Stuffing Attack, and Collusion Attack [5–8]. According to the behavioral characteristics of malicious nodes, they can be divided into individual and collusion malicious. These malicious behaviors destroy users' trust in P2P systems. Individual malicious nodes tend to act alone and make fault evaluation to any other nodes, while collusion malicious nodes act as a team causing more serious consequence.

In large scaled P2P networks, how to quickly calculate the mass trust evidences is a challenge to the efficiency of trust computing, and how to defend against collusion malicious behaviors is also a big challenge hindering the effectiveness of trust management. The main objective of this paper is to propose an iterative and dynamic trust computing model, named IDTrust for short, to effectively evaluate nodes trust degree against both individual and collusive malicious behaviors in

large scaled P2P systems. First of all, explicit and implicit evidences are defined in IDTrust and then direct trust, indirect trust, global trust, and decision trust are modeled in mathematical expressions based on evidences. This paper has the following innovative features.

(1) How to obtain large number of trust evidences and make metric decision iteratively and dynamically is the key in trust computing within an environment of large scale P2P Internet. As a result, we propose a kind of computing architecture with hierarchical distribution trust, which layers these three parts independently: evidence collection, trust computing, and P2P communication, leading to a three-layered P2P topology and enhancing the distribution computing of P2P commercial communication.

(2) Traditionally, each trust computing involves all the trust evidence, which contradicts the characteristic of P2P rapid and large scaled data. Therefore, in this paper, we propose an iterative and dynamic trust computing model. Within this model, trust computing is iterative and only requires the newest trust evidence. The performance of trust computing is better.

(3) This paper models and measures the trust evidence from two aspects, explicit feedback and implicit feedback of customer, to ensure maximum accuracy and integrity. We design direct trust, indirect trust, global trust, and decision trust in IDTrust against individual malicious and collusion node behaviors, to make trust management more effective.

In the remainder of the paper, we introduce some related works in the next section. Section 3 describes the trust computation architecture (named IDTrust-Arch) which provides iterative and dynamic trust computation in P2P network for IDTrust. Trust evidence modeling in IDTrust is proposed in Section 4, and trust measurement of IDTrust is listed in Section 5. The simulations and analysis follow in Section 6, with conclusions afterwards in the last section.

2. Related Work

Researchers have done lots of work around trust computing in decades. Most of these achievements are about trust issues in P2P network, distributed ad hoc, sensor network, Internet, and so forth. Trust computing includes three modules, namely, trust evidence acquisition, trust evaluation, and trust inference. We conclude current related work into four kinds in this section.

(1) Trust Computing Based on Local Transaction Evidences. Direct trust or local trust expresses to what extent a trustor believes a trustee based on the trustor's own local transaction history with the trustee. Buchegger and Le Boudec proposed CONFIDANT protocol based on local information [9], and the model processes secondhand information directly through friends in P2P network or ad hoc network. This method could defend against single malicious behaviors but could not avoid betrayal and collusion. Damiani et al. present local trust model named XRep [10]; they believe that nodes from same IP cluster are collusion group in P2P network. XRep has good convergence to recognize collusion behaviors, but it has limitation and some honest nodes may also be included in the collusion IP cluster. Jia et al. [11] optimize direct trust by power law and design forgetting factor based on time. Their model makes trust decision through a process of asking the neighbors' feedback of the target node. This model is effective but not versatile because it assumes that the nodes must have trusted neighbors. Direct trust has certain cognitive abilities on single malicious node, but it has limitations on node collusion and malicious tampering evidence.

(2) Trust Computing Based on Global Historical Information. Global trust is computed from other nodes recommendations to measure how trustworthy the target node is. To compute global trust, a trustor needs to aggregate all the trustee's trust evidences.

There are two major ways to calculate global trust. The first way is based on central server calculation, such as eBay which requires the peer nodes to deal with each other on the central platform. The other way is fully distributed, such as EigenTrust [12] which is proposed by Kamvar et al. EigenTrust reputation system can infer a unique global trust in a very distributed way by history. Such a global model does not need an administration center and difficultly guarantees a fast and secure convergence when computing the global trust. Nevertheless, it inspires our works. Dou et al. [13] improved the EigenTrust system in computing convergence and model security. However, there remains an efficiency problem and its security mechanism is only from punishment and certification. Xiong and Liu [14] proposed a PeerTrust model with three basic trust parameters and two adaptive factors and then defined a general trust metric to combine them efficiently. Jøsang et al. [15] proposed a trust inference method for simplifying a complex network to express it in a series of parallel networks. This solution may lead to the loss of trust information. They proposed an edge splitting method in the further works [16] to address this problem. Nevertheless, this method is valid only on a simple trust network and invalid on complex trust networks. Wang and Nakao propose Poisoned Water [17] and Shaikh et al. propose GTMS [18]; they measure trust based on global group to improve the convergence rate of the trust calculation and global trust accuracy. Song et al. propose global trust based on fuzzy logic inference rules [19]. This method has higher detection rate of malicious nodes, but it can only avoid simple malicious behaviors and cannot defend against various attacks on trust mechanism. Gan et al. [20] present a new method for trust recommendation. This method uses confidence factor to comprehend direct and recommendation trust and establishes reward and punishment model to encourage fair global recommendation.

(3) Trust Computing Based on Global and Historical Information Correlation. Correlation trust is based on global trust with similarity or correlation factors. Li et al. [21] propose a global trust SWRTrust with cosine similarity, which can identify collaborative cheating malicious behaviors. Das and Islam [22] propose an excellent dynamic trust computation

model SecuredTrust to cope with the strategically altering behavior of malicious agents and to distribute workload as evenly as possible among service providers. Qiao et al. [23] propose a novel network group behavioral model based on trust by exploring behavior similarity, aiming at perception and response of malicious network incidents. The proposed model establishes trust relationship between nodes using large scaled network topology and uses relevant trust concept to increase trust value between weak correlations. Other models like [24, 25] also propose global trust with correlation computation to improve trust decision performance based on large scaled transaction histories.

(4) Trust Computing Based on Inference Network with Multifactors. Another kind of trust computing model is based on inference network with probability or multidimensional factors. Kuter and Golbeck [26] propose a trust inference model SUNNY based on trust network by evaluating trust and confidence with inference probability. It has good effectiveness to discover trust recommender paths. Vu and Aberer [27] use trust model to monitor dishonest behaviors via filtering unfair ratings in trust network. Wang and Singh [28] propose trust model based on evidences conflict probability. And Can and Bhargava [29] propose a distributed trust computing algorithm SORT, where nodes create their own trust network according to local trust information and infer the trustworthiness of target nodes according to history interaction and recommendation information. Liu et al. [30] infer trust relationships by small world theory and apply the proposed model in service network. Tan et al. [31] propose a novel trust inference model based on probabilistic model and balance theory for P2P network, and the proposed algorithm could discover more valuable trust evidences paths for inferring the target's trust. They applied the proposed model in [32] and get a relatively effective inference performance. Gradually, researchers begin to infer trust degree with multidimensional evidence factors. Wang and Wu [33] proposed a multidimensional evidence-based trust management system with multitrusted paths (MeTrust for short) to conduct trust computation on any arbitrarily complex trusted graph. The trust computation in MeTrust has three tiers, namely, the node tier, the path tier, and the graph tier. It is an excellent trust model. However, it does not provide distributed storage structure for P2P system. Jiang and Li presented a novel reputation-based trust mechanism for P2P e-commerce systems [34]. In this mechanism, one peer has two kinds of reputations, local reputations and global reputations. To compute the local and global reputations precisely and to obtain stronger resistibility to attacks as well, they use many comprehensive factors in computing trust value in the mechanism. Basically, this model is a comprehensive mechanism. However, its time factor is only linear and there is no clear method to resist team malicious behaviors. Tan et al. [35] presented a global trust model with correlation factor based on communication history and improved the time factor with exponential equation. It shows a rational history vector and presents three trust models with multidimensional trust factors.

Inspired by the above four kinds of trust computing model, in this paper, the proposed IDTrust has direct trust, indirect trust, global trust, and decision trust. Meanwhile, the IDTrust aims to solve trust issues in large scaled P2P which have dynamic and rapid evidence and to improve trust computing efficiency in iterative and dynamic computation based on evidences with multifactors. Some definitions and parameters have been discussed in our former work [24, 31, 32, 35], and IDTrust integrates methods and improves computation architecture.

3. IDTrust-Arch

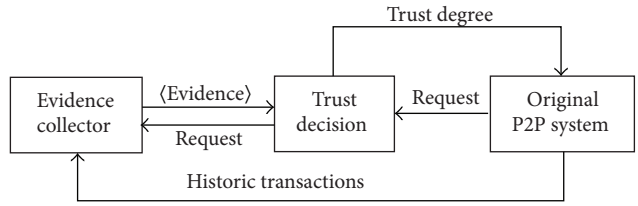
Trust measurement must be based on transaction or communication histories. Large scaled P2P systems have high frequency in generating data, strong dynamics in transactions and large scaled in transaction histories. As a result, trust evidences also have such same characteristics as high generating speed and large scaled histories. How to compute these dynamic and large scaled trust evidences efficiently becomes a challenge to trust model.

In IDTrust, we firstly design a distributed iterative computing architecture named IDTrust-Arch as shown in Figure 1(a), including *evidence collector* (EC) and *trust decision* (TD) which are designed extendedly based on P2P system architecture. In IDTrust-Arch, EC collects historic transactions from P2P system and feeds back evidence vector to TD, while TD computes the trustworthiness of objective nodes according to evidence vector and feeds back trust degree to P2P system according to request instructions. Relations of EC, TD, and original *P2P service* in IDTrust-Arch have the following features.

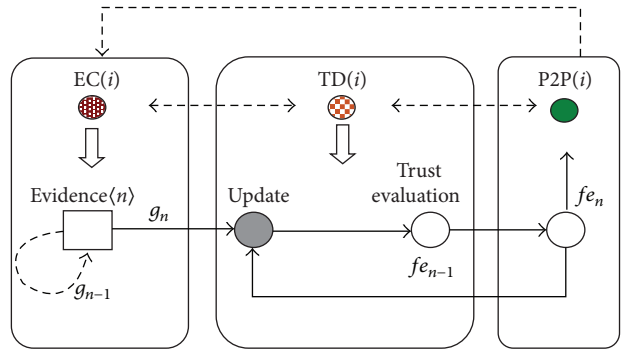
(1) Independent and Extended Functions. Within large scaled P2P system, distributed trust computing may cause huge communication traffic likely influencing the original P2P service quality. We mentioned this situation and it is under consideration in IDTrust-Arch. Different from traditional P2P trust model which treats trust computation as an inner-binding service, the proposed *IDTrust-Arch* is independent from P2P service. Firstly, EC, TD, and *P2P service* have separate communication ports, and EC and TD do not occupy P2P information channel. For any peer node, EC and TD are external extended functions but not internal binding service. Secondly, at any time, EC, TD, and *P2P service* can be deployed in different physical machines or different processes separately.

(2) Distributed but Topology Consistency. As a result of the independent functional design, each node i in P2P system can be extended into three nodes as node(i), EC(i), and TD(i). EC (and also TD) in all peer nodes can constitute a P2P network independently, using same look-up routing algorithm as the original P2P system. Figure 1(d) shows the topology relations between EC, TD, and the original P2P system. As the figure shows, the three will have the same topologies and communicate with each other.

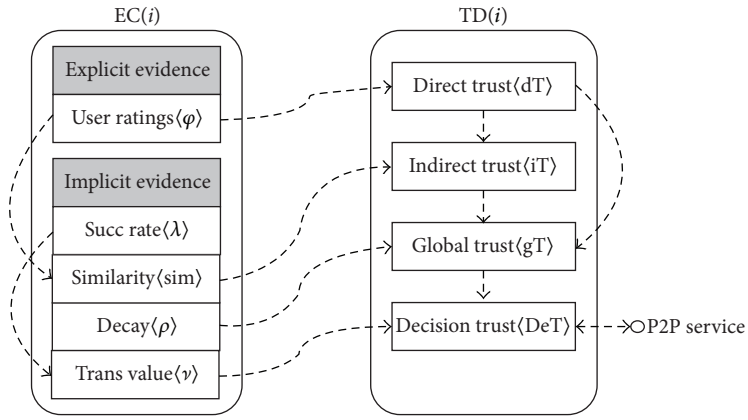
(3) Iterative Computing. IDTrust-Arch designs an iterative computing model for large scaled evidences in TD and EC, which only computes latest evidence vector each time based



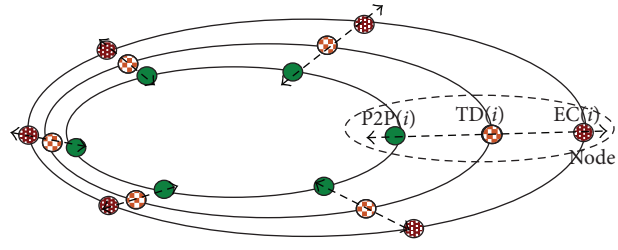
(a) Relations between EC, TD, and original P2P topology in IDTrust-Arch



(b) Iterative computing model in IDTrust-Arch



(c) Relations between evidences and trust model in IDTrust



(d) Topology

Figure 1: IDTrust-Arch.

on former result. Figure 1(b) shows the iterative trust computing processes. We can see that each computation is using the result of last computing. This kind of new method can greatly improve the trust computing efficiency and contains all the information of evidence by calculation of iterative accumulation, comparing it to the traditional one, which involves all the evidence every time. When large numbers of nodes in P2P network are online dynamically, for example, sometimes online, sometimes offline, traditional computing model results in difference of trust evaluation due to the inconsistence of set of historic evidences (e.g., there are 100 evidences of activity last time but may be only 30 next time). The iterative model in IDTrust-Arch can make use of evidences more efficiently by keeping evaluation consistency all the time. This

Table 1: Example: cumulative evaluation value of $\varphi_n(1,2)$.

Iteration number	$\varphi^{t_{\text{now}}}(1,2)$	Time stamp	α	$\varphi_n(1,2)$
Initial	Ratings from N1 to N2	$T1 = 10$	1	0
1	0.95	$T2 = 20$	1	0.95
2	0.9	$T3 = 30$	0.75	0.9125
3	0.7	$T4 = 80$	0.918367347	0.717346939
4	0.8	$T5 = 100$	0.395061728	0.75

kind of trust computation of IDTrust-Arch can adapt to high speed and large scaled P2P evidences flexibly. The following equation is the formalized formula of updating function:

$$\begin{aligned} fe_n(p,q) &= \text{update}\left(fe_{n-1}(p,q), g_n(p,q)\right), \\ g_n(p,q) &= \text{update}\left(g_{n-1}(p,q), \text{evidence}(p,q)\right). \end{aligned} \tag{1}$$

Here, we assume that node p (called trustor) needs to calculate the trust degree of node q (trustee). And $fe_n(p,q)$ denotes the nth trust decision computing in node p while $g_n(p,q)$ denotes the nth evidence modeling. Appropriate initialized data will be set for the very first iterative computing. The following IDTrust modeling is based on IDTrust-Arch and (1). It has been noted that (1) is only a formalized iterative computation method to guide modeling in IDTrust, and actual model symbols and parameters depend on model's requisitions.

4. Trust Evidence Modeling for IDTrust

Trust evidences in IDTrust come from EC, including explicit feedback evidence and implicit feedback evidence. The explicit feedback evidence is from customer nodes active evaluation, while implicit feedback evidence should be excavated from transactions. We formalize trust evidence vector in IDTrust as Evidence(p,q) to represent evidences that node p has upon node q. That is modeled by

$$\text{Evidence}(p,q) = \left\langle \langle \varphi \rangle, \langle \lambda, \nu, \rho, \text{sim} \rangle \right\rangle. \tag{2}$$

Explicit evidence vector in IDTrust is mainly consisting of user ratings φ, and of course it could be further expanded to multidimensional evidence vector in specific P2P systems. Implicit evidence vector in IDTrust mainly consists of transaction success rate λ, transaction value ν (based on transaction price), trust decay factor ρ, and nodes similarity sim. The following will model both explicit evidence vector $\langle \varphi \rangle$ and implicit evidence vector $\langle \lambda, \nu, \rho, \text{sim} \rangle$ in iterative style according to (1).

4.1. Explicit Evidence. Usually explicit evidence is fed back by user ratings or satisfactions. In practice, the quantitative or qualitative feedback can be by scores (5 points or 100 points) or by satisfaction (satisfied, mostly satisfied, not satisfied, etc.). In this paper, we use *user evaluation* by range $[0,1]$ to unify users' ratings or satisfactions.

Definition 1. User evaluation is denoted by $\varphi^t(p,q) \in [0,1]$ and represents that user p evaluated user q with a $[0,1]$ rating after transaction with q at the time t. $\varphi^t(p,q) = 0$ represents that node p is fully unsatisfied with node q's service, while $\varphi^t(p,q) = 1$ represents being completely satisfied. Let $\varphi^{t_{\text{now}}}(p,q)$ mean the latest user evaluation at the time t_{now}, while $t = t_1$ means the very beginning evaluation time.

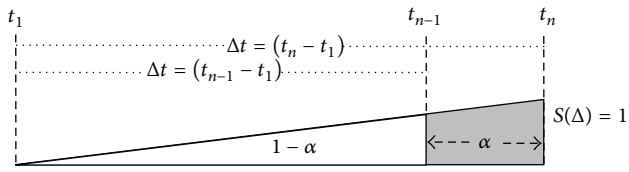


Figure 2: Dynamic factor of current evaluation.

Definition 2. Using $\varphi_n(p,q)$ to denote the current *cumulative evaluation value* by node p upon node q, based on the evaluation of last time according to iterative model, assume $\varphi_0(p,q) = 0$; that is,

$$\varphi_n(p,q) = \alpha \cdot \varphi^{t_{\text{now}}}(p,q) + (1-\alpha) \cdot \varphi_{n-1}(p,q), \tag{3}$$

where the weight α is the dynamic factor of current evaluation $\varphi^{t_{\text{now}}}(p,q)$. In IDTrust, α will be dynamically changed by the distance between t_{n-1} and t_n, and larger distance will bring bigger α, as shown in Figure 2. In other words, the current evaluation $\varphi^{t_{\text{now}}}(p,q)$ will be more important if user p did not transact with q after a longer time.

In Figure 2, we calculate α by the method of right triangle where area is 1 (unit area) and the base is global time difference (from t_1 to t_{now}) of interaction between p and q. Therefore, assume $\alpha = 1$ at the very beginning ($t = t_1$); the weight α of current evaluation is the area of the triangle from time t_{n-1} to t_n. That is,

$$\alpha = \begin{cases} 1 - \left(\dfrac{t_{n-1} - t_1}{t_n - t_1}\right)^2, & \text{if } t_n > t_1 \\ 1, & \text{if } t_n = t_1. \end{cases} \tag{4}$$

For example, assume there are four transactions that happened between node $N1$ and node $N2$, just as shown in Table 1. Then the cumulative evaluation value would be computed according to (3) based on evaluations from $N1$ to $N2$, while α changed based on transaction time stamps. After four iterations, the value of $\varphi_n(1,2)$ is 0.75. This value is less than the average of transaction rations from $N1$ to $N2$ because of the dynamic changes of α.

4.2. Implicit Evidences

4.2.1. Successful Rate. Successful transaction is a positive stimulus for nodes, while failure of transaction (or other situations such as complaint) would result in a negative growth to nodes trust.

Definition 3. $\lambda_n(p,q)$ denotes the percentage of *cumulative successful transaction* of node p to node q. Assume $\lambda_0(p,q) = 0$. That is,

$$\lambda_n(p,q) = \frac{(n-1)\,\lambda_{n-1}(p,q) + \lambda^{t_{\text{now}}}(p,q)}{n}, \tag{5}$$

where $\lambda^{t_{\text{now}}}(p,q) = \{0, \text{if successful}; 1, \text{if failed}\}$ represents whether the current nth transaction is successful or not.

4.2.2. Transaction Value. Transaction amount (in real or virtual currency) is typical implicit trust evidence. People usually think that the service provider with larger transaction amount is more valuable to trust in intuition. However, people are also worried about the malicious or dishonest block trade which may be only baits for honest users. Thus, we model the transaction amount as *transaction value* which is based on both transaction amount and successful rate.

Definition 4. Use $\nu^{t_{\text{now}}}(p,q) = \omega^{t_{\text{now}}}(p,q) \cdot \lambda_n(p,q)$ to denote the *current transaction value* of the current user transaction amount $\omega^{t_{\text{now}}}(p,q)$.

For instance, if the current transaction amount $\omega^{t_{\text{now}}}(p,q)$ is 100 but the current successful rate $\lambda_n(p,q)$ is only 0.25, then we take the current transaction value to be only $100 * 0.25 = 25$. This engagement can stimulate honest users to improve their transaction successful rate λ and also suppress the malicious behaviors which strategically raise the trustworthiness by transaction amounts meanwhile.

Definition 5. Using $\nu_n(p,q) = \nu^{t_{\text{now}}}(p,q) + \nu_{n-1}(p,q)$, to denote *cumulative transaction value* of the transaction amount, assume $\nu_0(p,q) = 0$.

4.2.3. Time Decay Factor for Trustworthiness. In the process of trust evaluation, the more recent the trust evidence is, the more important it is. If a user node takes a long time without transacting with other nodes, its trust will be degraded gradually. We apply exponential function to design the decay factor of direct trustworthiness of nodes, which will be used in direct trust in IDTrust. That is,

$$\rho = e^{-\kappa \Delta t}. \tag{6}$$

Here, k ($k > 0$) is the decay speed, and $\Delta t = t_n - t_{n-1}$ denotes the time difference for a certain user node. Figure 3 is the illustration of decay curves of four different decay speeds. Apparently, the lager k is, the faster the decay would be.

4.2.4. Similarity of User Evaluations. Similarity is one kind of important implicit evidence which indicates the extent to which two nodes are alike, especially when there is no direct transaction between two strange nodes. In order to measure the similarity between user nodes p and q, we choose the third set of nodes that have interaction with both of them to be a computing object. For example, node A did not know B, but both share common friends $\{C, D\}$, and then A can judge similar correlation with B via its friends $\{C, D\}$. If the communication history among $A \sim \{C, D\}$ is similar to the history among $B \sim \{C, D\}$, then we think A and B have very similar correlation.

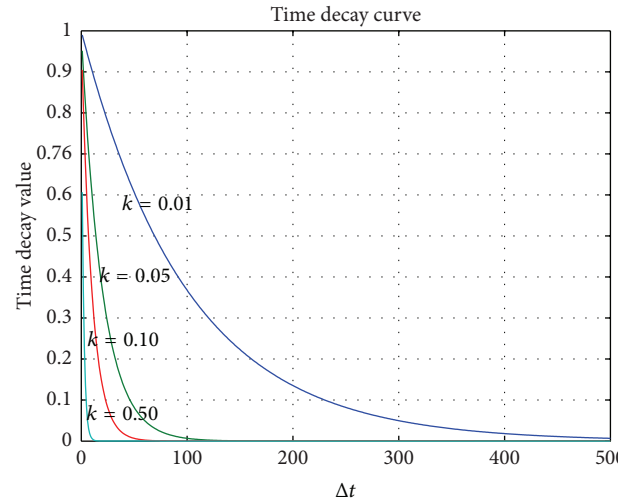


Figure 3: Time decay curves.

Let I_x denote the set of nodes which node x has ever interacted with. Let $I_{pq} = I_p \cap I_q$ denote the common third-part set which interacts with both users p and q. The similarity computation is based on *user-user* matrix $R(n \times n)$ whose element r_{ij} is

$$r_{ij} = \begin{cases} \varphi_n(i,j), & \text{if } i \neq j \\ 1, & \text{if } i = j. \end{cases} \tag{7}$$

If no user interaction happened between user nodes i and j, we assume $r_{ij} = 0$. Then,

$$R = \left(r_{ij}\right)_{n \times n} = \begin{bmatrix} 1 & r_{12} & \cdots & r_{1n} \\ r_{21} & 1 & \cdots & r_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ r_{n1} & r_{n2} & \cdots & 1 \end{bmatrix}. \tag{8}$$

Each row i in R denotes evaluation to all other user nodes from node i. There are many methods to calculate the similarity between two items, such as cosine similarity, correlation similarity, and adjusted cosine similarity. Cosine similarity is a general method to compute the similarity between two user nodes. However, pure cosine formula cannot distinguish the subjective difference of evaluation criteria that different users have for the same object. For instance, for a certain user x, user p evaluated x with value 0.9 after a transaction, and user q evaluated it with value 0.6 in another transaction. The evaluated value is quite different; however, it is probable that

0.9 and 0.6 both stand for "*satisfied*" separately in perspective of p and q, and the difference may only be because user q has more strict evaluation criteria. In this situation, ordinary cosine formula is not appropriate. Thus, we apply the adjusted cosine formula that is subtracted by average value to compute the similarity based on matrix R. That is,

$$c\,\mathrm{sim}_n(p,q) = \frac{\sum_{c\in I_{pq}}\left(\varphi_n(p,c)-\overline{\varphi_n(c)}\right)\cdot\left(\varphi_n(q,c)-\overline{\varphi_n(c)}\right)}{\sqrt{\sum_{c\in I_{pq}}\left(\varphi_n(p,c)-\overline{\varphi_n(c)}\right)^2}\cdot\sqrt{\sum_{c\in I_{pq}}\left(\varphi_n(q,c)-\overline{\varphi_n(c)}\right)^2}}, \quad (9)$$

where $\overline{\varphi_n(c)}$ denotes the average evaluation value of node $c \in I_{pq}$.

In the above similarity equation, three questions need to be further considered.

(1) Sparse Common Nodes. It is a challenge when $|I_{pq}|$ is zero or very small. This phenomenon may even result in similarity beyond computing. Thus, in IDTrust, we assume that similarity should be computed by cosine only when $|I_{pq}|$ reaches threshold τ (natural number that is larger than 1).

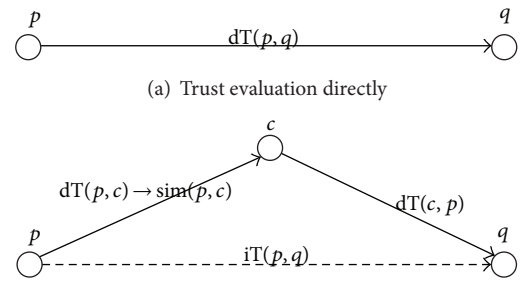
(2) Result Adjusting. Although the range of user evaluation φ is within [0, 1] in IDTrust, due to inner product of deviation, the return value of adjusted cosine similarity by (9) is within $[-1, 1]$, where range $[-1, 0]$ denotes the evaluation vectors angle of p and q is between degrees [90, 180] which means the two vectors are completely irrelevant (orthogonal) or even opposite. Thus, we assume it is completely dissimilarity in this case, and the similarity is set to zero when $c\,\mathrm{sim}_n(p,q)$ return $[-1, 0]$ in IDTrust. As a result, the range of similarity will be adjusted to [0, 1], which is easier to control.

(3) Default Similarity. At the very beginning (or when similarity is beyond computing), we need to initialize a default similarity. In IDTrust, the default similarity is 0.5.

Hence, (9) is modified to be

$$\mathrm{sim}_n(p,q) = \begin{cases} 0, & \text{if } \left(\left|I_{pq}\right| > \tau\right) \wedge \left(c\,\mathrm{sim}_n(p,q) \in [-1,0)\right) \\ c\,\mathrm{sim}_n(p,q), & \text{elseif } \left(\left|I_{pq}\right| > \tau\right) \wedge \left(c\,\mathrm{sim}_n(p,q) \in [0,1]\right) \\ 0.5, & \text{otherwise.} \end{cases} \quad (10)$$

Also, we assume the similarity of $\mathrm{sim}_n(x,x) = 1$, which means the similarity among nodes is reflexive. Note that similarity among nodes is symmetrical; that is, $\mathrm{sim}_n(p,q) = \mathrm{sim}_n(q,p)$. Therefore, similarity among nodes denotes the compatibility relationship. It indicates that we can get a similar set in the whole set by compatibility relationship, which may be a new way to get complete coverage for similar nodes in the whole set in our further research. But in IDTrust now we have not applied this feature.



(a) Trust evaluation directly

(b) Trust evaluation indirectly

Figure 4: Direct/indirect trust computing.

5. Iterative Trust Measurement Models in IDTrust

Based on the above iterative computing architecture and evidence modeling, we design direct trust model, indirect trust model, global trust model, and decision trust model for IDTrust, against malicious behaviors especially collusion behaviors. Figure 1(c) shows relations between trust evidences and trust measurement models.

5.1. Direct Trust. Direct trust (also called local trust) is based on the direct transaction experiences between the trustor and trustee, as Figure 4(a) shows. We define the direct trustworthiness of node p to node q from user evaluation which is from the direct transaction between the two nodes. Let $\mathrm{dT}_n(p,q) \in [0, 1]$ denote the direct trust of node p to node q. That is,

$$\mathrm{dT}_n(p,q) = \varphi_n(p,q). \quad (11)$$

As you can see, the better the service that node q provided, the higher the direct trust degree that node q obtains from node p.

5.2. Indirect Trustworthiness. To compute indirect trust by traditional trust models, the trustor needs to aggregate all of the trustee's transaction evaluations from recommenders and then compute or infer indirect trust degree of target node by those recommenders' evidences. Figure 4(b) illustrates that node p calculates indirect trust degree of node q through recommender node c.

As we know, we can obtain the direct trust $\mathrm{dT}(c,q)$ from recommender node c and direct trust $\mathrm{dT}(p,c)$ from p. Some related indirect trust models use these two kinds of direct trust to calculate indirect trust of the target node, such as $\mathrm{dT}(p,c) * \mathrm{dT}(c,q)$ or other similar forms. However, since the direct trust is usually based on user evaluation and the user evaluation is subjective in a certain extent, different users have different evaluation criteria, just as we discussed in Section 4.2.4 about similarity of user evaluations above. Thus, we consider user similarity between nodes p and c instead of direct trust of p and c, for the similarity computation in IDTrust is also based on user evaluation (which is also equivalent to direct trust). Using similarity can also avoid collusion

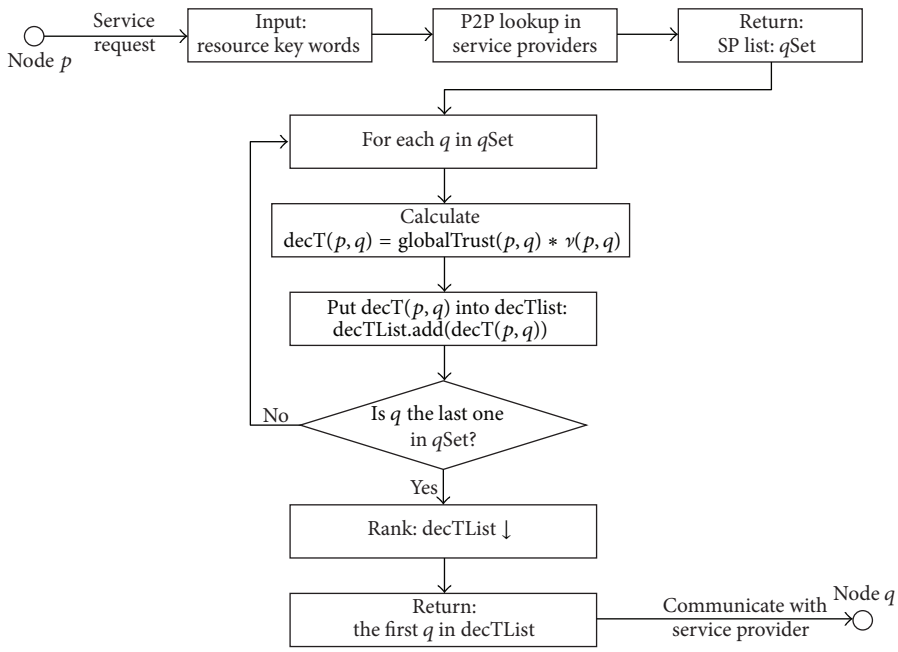


Figure 5: Trust decision process.

malicious recommenders, since malicious evaluation would quite differ from the honest evaluation.

Let $\mathrm{iT}_n(p,q) \in [0,1]$ denote indirect trust of node p to node q, which is calculated by direct trust of recommender set cSet to node q, and use the similarity between node p and cSet as another indirect parameter. That is,

$$\mathrm{iT}_n(p,q) = \begin{cases} \sqrt{\dfrac{\sum_{c \in c\mathrm{Set}} \mathrm{sim}_n(p,c) \cdot \mathrm{dT}_n(c,q)}{|c\mathrm{Set}|}}, & \text{if } |c\mathrm{Set}| > 0 \\ 0, & \text{otherwise,} \end{cases} \quad (12)$$

where if recommender nodes set cSet is empty, the indirect trust return is zero. In order to read the result clearly, the above formula amplifies the value with square root, since both similarity and direct trust ranged within [0, 1].

5.3. Global Trust. Now, we define the global trust as the weighted sum of direct trust and indirect trust, where the general weight is the decay factor ρ:

$$\mathrm{gT}_n(p,q) = \rho \cdot (\zeta \cdot \mathrm{dT}_n(p,q) + (1-\zeta) \cdot \mathrm{iT}_n(p,q)) + (1-\rho) \cdot \mathrm{gT}_{n-1}(p,q). \quad (13)$$

Here, $\zeta \in [0,1]$ is the weight of the direct trust with default value 0.5 which could be adjusted by users according to the requirement.

5.4. Decision Trust. Decision trust is an integrated value built from iterative global trust and implicit evidence of transaction value, to represent a comprehensive expression for user nodes' trust.

Let $\mathrm{decT}_n(p,q) = \mathrm{gT}_n(p,q) \cdot \nu_n(p,q)$ denote decision trust which is based on global trust and implicit evidence transaction value. For example, if $\mathrm{gT}_{1010}(1,3) = 0.65$ and transaction value $\nu_{1010}(1,3) = 100$, we can get $\mathrm{decT}_{1010}(1,3) = 65$ indicating the integrated trust of node 3 is 65 from the perspective of node 1. As we can see, the $\mathrm{gT}_n(p,q)$ is a global factor while $\nu_n(p,q)$ is a relatively local factor; we apply these two factors to help in decision making.

Hereby, we also design a trust decision algorithm as shown in Figure 5. In the algorithm process, as a trustor, user node p initiates the trust decision in order to find the most trustworthy nodes. At the beginning, node p inputs resource keywords to request P2P services, and then P2P network returns a service provider list (qSet) specifying who can provide related resource services. Each decision trust (decT) will be calculated through EC and TD, and all the decision trust will be ranked in descending order. At last, the node q that has the highest decision trust value will be recommended to node p.

6. Simulations and Analysis

In order to prove the rightness and efficiency of IDTrust, we design a simulation system based on IDTrust-Arch and our former work. We will separately verify IDTrust performance against dynamic individual malicious and collusion malicious environment and verify the iterative computation performance comparing with traditional model.

6.1. Simulation Environment. Firstly, we design a P2P communication architecture as in Figure 6 based on IDTrust-Arch introduced above. In the system, service provider (SP)

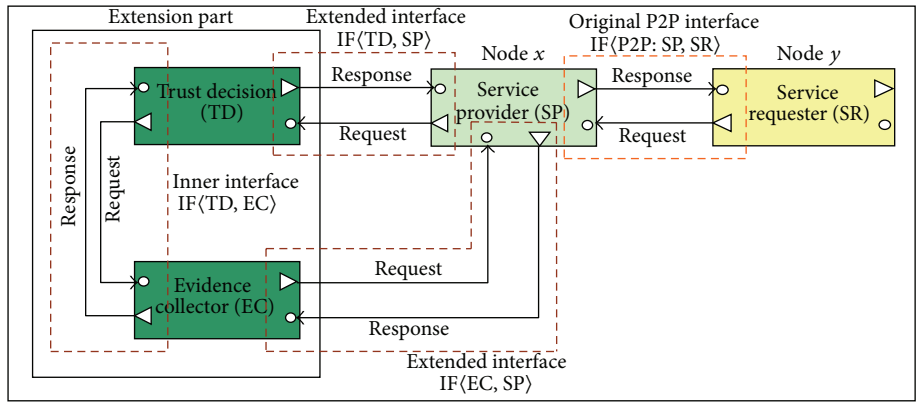


Figure 6: Communication architecture of simulation system.

is the node that provides services while service requester (SR) needs the resource services. EC is to collect and integrate the historic data and obtain evidence vector from P2P system. TD is to compute the trustworthiness of objective nodes according to evidence vector provided by EC. The original P2P interface IF⟨P2P: SP, SR⟩ manages the communication between node x and node y. The extended interface IF⟨TD, SP⟩ manages the communication between SP and TD. And interface IF⟨EC, SP⟩ manages the communication between SP and EC. The inner interface IF⟨TD, EC⟩ manages the communication between TD and EC.

We implement the above simulation system based on MS.NET Framework 4.5 and MS P2P Networking platform with *c#* programming language, thread pool technique, and distributed communication. In topology, we use simplified *Chord* to be the P2P routing algorithm.

We simulate three kinds of P2P nodes similar to [24, 35].

(1) Node of Class A. It is the normal node in P2P system, which provides normal service and evaluates service almost equal to simulated resource value.

(2) Node of Class B. It is an individual malicious node that provides false service and makes fault evaluation for any other nodes at random independently.

(3) Node of Class C. It belongs to some malicious team nodes, providing false service, and evaluates normal nodes with fault feedback. It should be noted that it overstates team members' services with high score to cheat other nodes out of their team.

We simulated 1000 nodes and each node has 50~100 resources abstract and designed 4 kinds of experiment and almost 1,000,000 simulated transactions happened between nodes. Table 2 shows the nodes proportion allocation for different experiments.

Table 3 initializes different parameters' value. Most of these parameters will be dynamically changed with the iterative computation going on.

Table 2: Nodes proportion setup in different experiments.

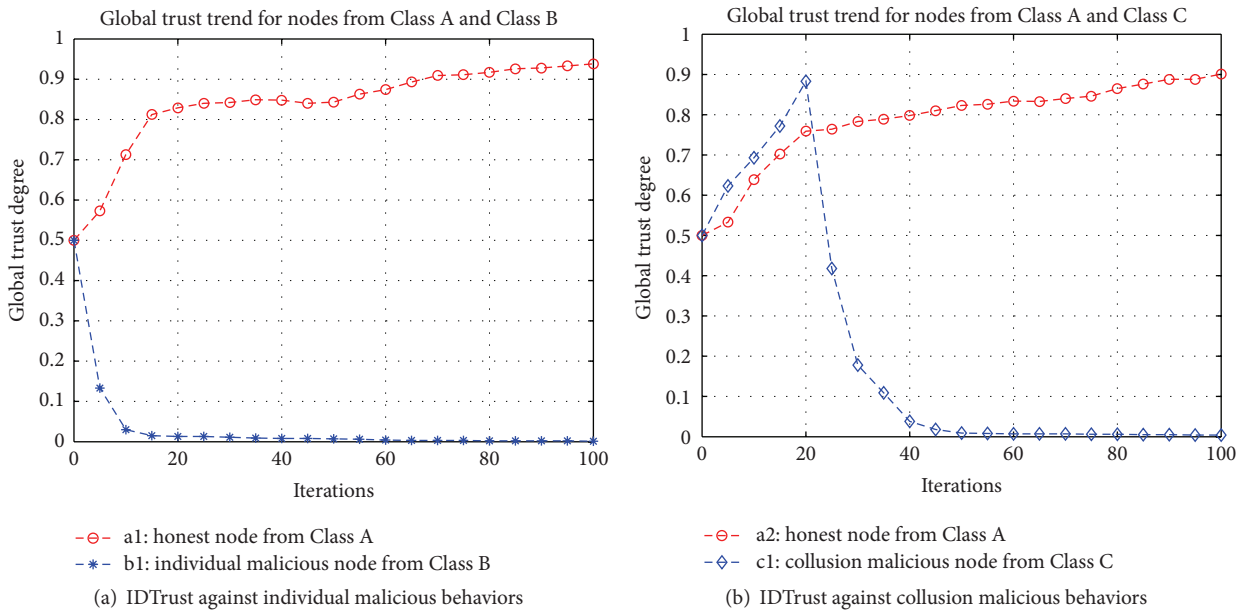
Experiment	Parameter	Allocation
Experiment 1	A node	75%
IDTrust against individual malicious	B node	25%
behaviors	C node	0%
Experiment 2	A node	75%
IDTrust against collusion malicious	B node	0%
behaviors	C node	25%
Experiment 3	A node	70%
Iterative performance	B node	15%
Experiment 4	C node	15%
Trust decision performance		

Table 3: Initialized parameters.

Parameter	Initialized value	Description
N	1000	The number of simulated nodes
φ_0	0	Initialized value of user evaluation
λ_0	0	Initialized value of successful transaction rate
ν_0	0	Initialized value of transaction value
sim_0	0.5	Initialized value of similarity
iT_0	0	Initialized value of indirect trust value
gT_0	0.5	Initialized value of global trust value
α	1	Initialized value of user's current evaluation
k	0.01	Decay speed
τ	10	Threshold of $\lvert I_{pq} \rvert$
ζ	0.5	Weight of direct trust which could be adjusted by users according to requirement

6.2. Simulation and Analysis

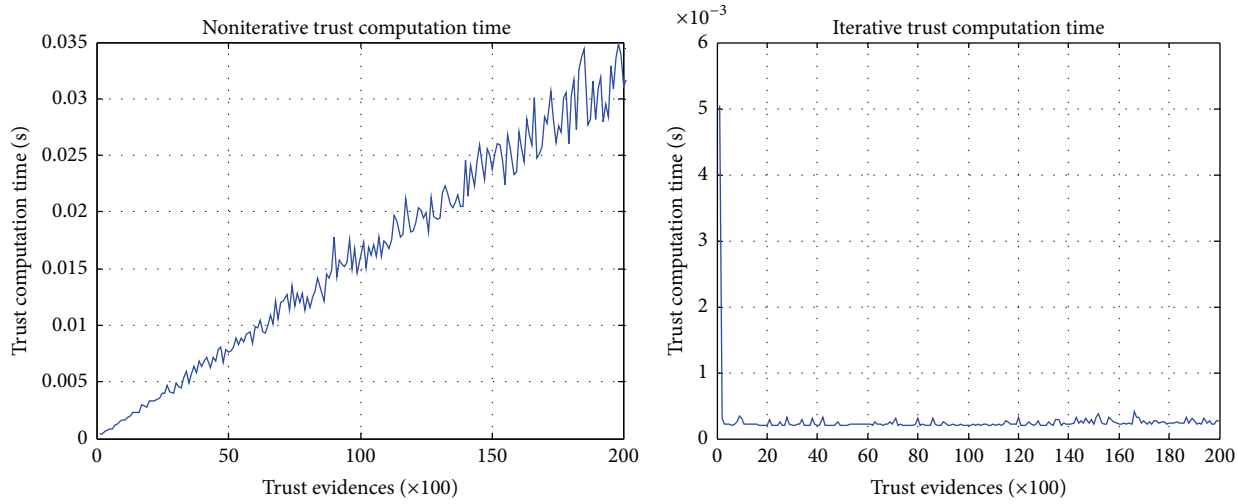
Experiment 1 (IDTrust against individual malicious behaviors). The first adversary model for IDTrust is individual malicious node. We set up almost 25% of individual malicious



(a) IDTrust against individual malicious behaviors

(b) IDTrust against collusion malicious behaviors

Figure 7: Simulation for defending against malicious attacks.



(a) Noniterative trust computation time

(b) Iterative trust computation time

Figure 8: Iterative computation performance comparing with noniterative computation.

nodes of Class B and 75% of normal nodes. From observation of IDTrust, global trust value of designed honest node of Class A and individual malicious node of Class B during simulation experiment, we can get result as shown in Figure 7(a). With increasing of iterative time, global trust value of observed individual malicious node decays very fast; in contrast, trust value of honest node increases all the way. This means that defense to individual malicious node is effective.

Experiment 2 (IDTrust against collusion malicious behaviors). In this experiment, we set up 25% of collusion malicious nodes of Class C and 75% of normal nodes. We want to see whether IDTrust can defend against collusion malicious behaviors. Figure 7(b) shows results. At the beginning, IDTrust cannot distinguish and recognize collusion malicious node due to historic set of public transactions. With iterative calculation increasing, the trust value of collusion node degraded drastically to almost zero. It indicates the similarity in IDTrust works well and IDTrust is effective to defend collusion behaviors.

Experiment 3 (performance of iterative calculation). In order to verify the efficiency of iterative calculation in IDTrust, we count the computation time cost of decision trust computation statistically in both iterative and noniterative calculations separately. In this experiment, we set up 15% of collusion malicious nodes, 15% of individual malicious nodes, and 70% of normal nodes. Figure 8(a) shows the result of noniterative

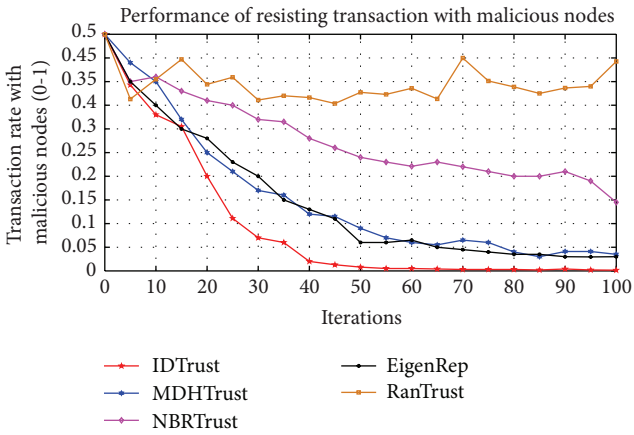


Figure 9: Trust decision performance comparisons.

calculation time cost while Figure 8(b) is about iterative calculation in IDTrust. As we can see from the comparison, the IDTrust with iterative calculation improves computation performance greatly, and its time cost is much smaller than that of noniterative one. The experiment indicates that architecture of IDTrust is highly efficient.

Nevertheless, this iterative performance still occupied CPU and memory utilization. During the simulation, the CPU (Intel Core i7-4980HQ@2.8 GHz) utilization is almost 60% and the memory (4 G DDR3 RAM) utilization is 73% or so. But this disadvantage would be helpful for our further study.

Experiment 4 (performance of trust decision). Finally, in this paper, we compare performance of trust decision of IDTrust with EigenTrust [12], MDHTrust [35], NBRTrust [24], and random model with same nodes proportion as Experiment 3. In this experiment, we count the number of malicious nodes of Class B and Class C which are selected to be trustees by normal nodes of Class A. Also, we define rejectRate to denote to what extent the normal nodes reject services against malicious nodes:

$$\text{rejectRate} = \frac{\text{Count of transacted malicious nodes}}{\text{Count of all transacted nodes}}. \quad (14)$$

Experiment result is shown in Figure 9. As we can see, IDTrust has rapid convergence to reject transactions with malicious nodes since it has strict trust decision processes as described in Section 5.4. The random trust model has relatively lowest performance for it has no trust decision definition. This experiment indicates that IDTrust is very sensitive to malicious node and has an advantage of recognizing malicious behavior.

7. Conclusions

In this paper, we propose a novel dynamic trust model with iterative computation for P2P systems, according to dynamics and fast responding characteristics in large scaled P2P network, named IDTrust.

Firstly, we propose a distributed computing architecture named IDTrust-Arch to obtain trust evidences and make trust decision, including three computing modules, EC, TC, and original P2P module, which are independent but communicate with each other, to get higher computation performance. This architecture separates the trust computing task from P2P system, while traditional trust computing is a binding service and based on universal set of evidence facts to obtain global trust degree of a given node.

Secondly, we propose an iterative computation method to model trust evidence vector and calculate trust in IDTrust. This design degrades the trust computation cost and improves the computation performance. Based on the above, IDTrust is proposed. Trust evidence vector in IDTrust includes both explicit and implicit trust evidences to improve the evidence integrity. Direct trust, indirect trust, global trust, and decision trust are designed in IDTrust based on explicit transaction evaluation and implicit evidence like transaction value, successful rate, decay factor, and similarity factor.

Finally, we design a simulation system based on IDTrust-Arch. Simulations against both individual and collusion malicious nodes prove that the IDTrust is right and efficient against the two adversary models. Simulations about iterative computation and trust decision prove that IDTrust has high computation performance and more efficient trust decision performance.

Conflict of Interests

The authors of this paper declare that they have no conflict of interests.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant no. 61402097 and no. 61572123; the National Science Foundation for Distinguished Young Scholars of China under Grant no. 61225012 and no. 71325002; the Specialized Research Fund of the Doctoral Program of Higher Education for the Priority Development Areas under Grant no. 20120042130003. The authors also acknowledge Ayebazibwe Lorretta's help in doing the grammatical revision.

References

[1] H. Esaki, "A consideration on R&D direction for future Internet architecture," *International Journal of Communication Systems*, vol. 23, no. 6-7, pp. 694–707, 2010.

[2] J. Duarte, S. Siegel, and L. Young, "Trust and credit: the role of appearance in peer-to-peer lending," *Review of Financial Studies*, vol. 25, no. 8, pp. 2455–2484, 2012.

[3] R. Emekter, Y. Tu, B. Jirasakuldech, and M. Lu, "Evaluating credit risk and loan performance in online Peer-to-Peer (P2P) lending," *Applied Economics*, vol. 47, no. 1, pp. 54–70, 2014.

[4] E. Lee and B. Lee, "Herding behavior in online P2P lending: an empirical investigation," *Electronic Commerce Research and Applications*, vol. 11, no. 5, pp. 495–503, 2012.

[5] Z. Jie, "A survey on trust management for VANETs," in *Proceedings of the 25th International Conference on Advanced Information Networking and Applications*, pp. 105–112, Singapore, March 2011.

[6] L. Mekouar, Y. Iraqi, and R. Boutaba, "Peer-to-peer's most wanted: malicious peers," *Computer Networks*, vol. 50, no. 4, pp. 545–562, 2006.

[7] H. Q. Lin, Z. T. Li, and Q. F. Huang, "Multifactor hierarchical fuzzy trust evaluation on peer-to-peer networks," *Peer-to-Peer Networking and Applications*, vol. 4, no. 4, pp. 376–390, 2011.

[8] C. Selvaraj and S. Anand, "A survey on security issues of reputation management systems for peer-to-peer networks," *Computer Science Review*, vol. 6, no. 4, pp. 145–160, 2012.

[9] S. Buchegger and J. Y. Le Boudec, "Performance analysis of the CONFIDANT protocol," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '02)*, pp. 226–236, New York, NY, USA, June 2002.

[10] E. Damiani, S. D. C. Vimercati, and S. Paraboschi, "A reputation-based approach for choosing reliable resources in peer-to-peer networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 207–216, ACM Press, Washington, DC, USA, November 2002.

[11] C. Jia, L. Xie, X. Gan, W. Liu, and Z. Han, "A trust and reputation model considering overall peer consulting distribution," *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 42, no. 1, pp. 164–177, 2012.

[12] S. D. Kamvar, M. T. 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 (WWW '03)*, pp. 640–651, ACM, May 2003.

[13] W. Dou, H. M. Wang, and Y. Jia, "A recommendation-based peer-2-peer trust model," *Journal of Software*, vol. 15, no. 4, pp. 571–583, 2004.

[14] 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.

[15] A. Jøsang, R. Hayward, and S. Pope, "Trust network analysis with subjective logic," in *Proceedings of the 29th Australasian Computer Science Conference (ACSC '06)*, vol. 48, pp. 85–94, Hobart, Australia, January 2006.

[16] A. Jøsang and T. Bhuiyan, "Optimal trust network analysis with subjective logic," in *Proceedings of the 2nd International Conference on Emerging Security Information, Systems and Technologies (SECURWARE '08)*, pp. 179–184, Cap Esterel, France, August 2008.

[17] 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.

[18] R. A. Shaikh, H. Jameel, B. J. d'Auriol, H. Lee, S. Lee, and Y.-J. Song, "Group-based trust management scheme for clustered wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 11, pp. 1698–1712, 2009.

[19] S. Song, K. Hwang, R. F. Zhou, and Y.-K. Kwok, "Trusted P2P transactions with fuzzy reputation aggregation," *IEEE Internet Computing*, vol. 9, no. 6, pp. 24–34, 2005.

[20] Z.-B. Gan, Q. Ding, K. Li, and G.-Q. Xiao, "Reputation-based multi-dimensional trust algorithm," *Journal of Software*, vol. 22, no. 10, pp. 2401–2411, 2011.

[21] J.-T. Li, Y.-N. Jing, X.-C. Xiao, X.-P. Wang, and G.-D. Zhang, "A trust model based on similarity-weighted recommendation for P2P environments," *Journal of Software*, vol. 18, no. 1, pp. 157–167, 2007.

[22] A. Das and M. M. Islam, "SecuredTrust: a dynamic trust computation model for secured communication in multiagent systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 2, pp. 261–274, 2012.

[23] L. Qiao, H. Hui, F. Bingxing, Z. Hongli, and W. Yashan, "Awareness of the network group anomalous behaviors based on network trust," *Chinese Journal of Computers*, vol. 37, no. 1, pp. 1–14, 2014.

[24] Z. Tan, H. Wang, W. Cheng, and G. Chang, "A distributed trust model for P2P overlay networks based on correlativity of communication history," *Journal of Northeastern University (Natural Science)*, vol. 30, no. 9, pp. 1245–1248, 2009.

[25] J. Yin, Z.-S. Wang, Q. Li, and W.-J. Su, "Personalized recommendation based on large-scale implicit feedback," *Journal of Software*, vol. 25, no. 9, pp. 1953–1966, 2014.

[26] U. Kuter and J. Golbeck, "Using probabilistic confidence models for trust inference in web-based social networks," *ACM Transactions on Internet Technology*, vol. 10, no. 2, article 8, 23 pages, 2010.

[27] L. Vu and K. Aberer, "Effective usage of computational trust models in rational environments," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 6, no. 4, article 24, 25 pages, 2011.

[28] Y. Wang and M. P. Singh, "Evidence-based trust: a mathematical model geared for multi agent systems," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 5, no. 4, article 14, 28 pages, 2010.

[29] A. B. 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.

[30] F. M. Liu, L. Wang, L. Gao, H. X. Li, H. F. Zhao, and S. K. Men, "A Web Service trust evaluation model based on small-world networks," *Knowledge-Based Systems*, vol. 57, pp. 161–167, 2014.

[31] Z. Tan, L. Zhang, and G. Yang, "BPTrust: a novel trust inference probabilistic model based on balance theory for peer-to-peer networks," *Elektronika ir Elektrotechnika*, vol. 18, no. 10, pp. 77–80, 2012.

[32] Z. H. Tan, G. M. Yang, and W. Cheng, "Distributed trust inference model based on probability and balance theory for peer-to-peer systems," *International Journal on Smart Sensing and Intelligent Systems*, vol. 5, no. 4, pp. 1063–1080, 2012.

[33] G. Wang and J. Wu, "Multi-dimensional evidence-based trust management with multi-trusted paths," *Future Generation Computer Systems*, vol. 27, no. 5, pp. 529–538, 2011.

[34] S.-X. Jiang and J.-Z. Li, "A reputation-based trust mechanism for P2P e-commerce systems," *Journal of Software*, vol. 18, no. 10, pp. 2551–2563, 2007.

[35] Z.-H. Tan, X.-W. Wang, W. Cheng, G.-R. Chang, and Z.-L. Zhu, "A distributed trust model for peer-to-peer networks based on multi-dimension-history vector," *Chinese Journal of Computers*, vol. 33, no. 9, pp. 1725–1735, 2010.