J. Basic. Appl. Sci. Res., 3(8)300-310, 2013

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

DTM: An Efficient and Dynamic Trust and Reputation Model for E-commerce Agents

Maryam Asadzadeh Kaljahi[1], Ali Payandeh[2], Mohammad Bagher Ghaznavi Ghoshchi[3]

[1]Department of IT, Tehran University, Iran
[2]Department of Computer Engineering Maleke- Ashtar University, Thehran, Iran
[3]School of Engineering, Shahed University, Tehran, Iran

ABSTRACT

In widespread and decentralized environment of the e-commerce clients are exposed to significant risk due to frequent interaction with unfamiliar, diverse, multi-agent and untrustworthy servers. As a consequence trust and reputation-based systems have been important issues and studied as an alternative to traditional security mechanism to reduce the risk. Different models of trust have been proposed to assess the credibility of agents, but they face the challenges of detecting and resisting dishonest and malicious advisors which collect local reputation scores and aggregate them into the total trust. This paper presents DTM- a dynamic and robust trust model which takes advantage of mostly previous model to trade off the damage of fake reputation information. The DTM model uses normal distribution factor for weight of recommendations and both number of iterations and fluctuating behavior factor on the dynamic adaptation between current and previous experiences. As a result of the aggregated trust will filter malicious behavior and reflect more accurate trust value. The DTM model can also resist against several attacks such as strategic and collusion attacks. Simulation experiments show that the DTM model can discern a small difference between real quality of service and computed trust, and the most notable is robust to malicious agents.
KEYWORDS*:* Trust, Reputation, E-commerce, Credibility, Malicious Attack, Security.

INTRODUCTION

In e-commerce clients often interact and take decisions under uncertainty with servers that are unknown to them and they are vulnerable to risk and have to manage it involved with the transactions. It is hard to solve these problems efficiently by conventional security policies, such as authentication [1,2,3] or security protocols such as SSL/TLS that has drawbacks against malicious servers[2,3]. One way to address this problem is trust-based approaches that can assist clients in accessing the level of trust they should place on a transaction. Trust is critical in such setting as it can make social interactions much fruitful as possible [4] and improve the robustness of the system. In the context of the e-services, trust is defined as: ‘‘Trust is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such an action and in a context in which it affects his own action [5]”. Trust is defined in term of evidence of future behavior based on interactions include direct measures and indirect measure as the reputation.

Like the interpersonal relationships in social networks, there are two kinds of trust between clients and servers: direct trust and recommendation trust (reputation). Direct trust means two agents (clients and server) directly exchange information and the reputation is based on collecting and aggregates recommendations about server’s past behavior. Reputation assumes a very important role when the client doesn’t have a sufficient knowledge of the server, so client can decide if server is reliable interlocutor or not.

But in e-commerce competitive environment agents can misbehave in a number of ways, such as providing fake recommendations on servers or showing a misleading, deceptive and malicious behavior to create problems to its competitors. The challenge of building a trust mechanism is how to effectively cope with such malicious behavior [6]. To suppress some deficiencies in existing trust models a novel, dynamic and robust trust model is introduced to decentralize e-commerce systems called DTM (Dynamic Trust Model).

The major contributions of this paper are illustrated as follows: filtering of unfair suggestions and establishing dynamic parameter to update trust measures. In DTM model weights of advisors are adjustable based on normal distribution factor to be able to efficiently distinguish reliable from deceptive feedbacks. As second contributions of DTM trust model, a new framework is introduced that determines a dynamic criterion to suitably provide and update the weights that should be assigned to the current with respect to previous experiences. In this way the accuracy of computed trustworthy will be improved. The remainder of this paper is organized as follows: In section

* **Corresponding Authors:** Maryam Asadzadeh Kaljahi, Department of IT, Tehran University, Iran
Email: mnewm22@yahoo.com

2, we present a summary of related work on this topic. The proposed trust model is presented with details in section 3. Experiments and simulation results are in section 4 and finally section 5 concludes the paper.

1. Related Works

So far, a very large variety of trust models is put forward to avoid deceptions, reduce risk and selfish agent's behavior in competitive communities like multi agent system (MAS), P2P systems, mobile agent systems, e-commerce [6,7,8], mobile ad hoc network (MANET) [19] and Vehicular ad hoc networks (VANETs) [20]. Which mostly information derived by direct experience and recommendations of other (reputation) can be used to compute total trust measure as decision criteria. However, among large-scale and loosely connected environments such as e-commerce direct experience is often not sufficient. In this case, prediction is mainly based on reputation. So trust models employ various strategies to distinguish unfair opinions. In the following sections, some of them are discussed. From now on, for the convenience of referring agents, the agent evaluating the trustworthiness of another is called the trustor and the agent being evaluated by trustor, is trustee.

The model defined by Marsh [9] is one of the basic models in the MAS system. It exploits trust upon three components base on basic trust, general trust and situational trust. The final decision is made through a well-defined threshold, but the schema doesn't verify the reputation of trustee.

A well-known trust mechanism is Eigen trust [11] that is applied also over P2P networks. It aggregates both types of direct and indirect trust and exploits distributed hash tables and pre-trust peers to compute a precise reputation and deal with unfair opinions. Eigen trust may encounter authorization restraints and in a large-scale P2P system it costs too much for any of the trustors to an afford [13]. Moreover, another challenge in competitive environments is how to discover pre-trust without centralized management.

The work described in [7] named Trummar, which is a comprehensive model that mobile agent systems can use in order to more resistant against malicious behaviors. This model estimates reputation of trustee by identifying three different types of advisors who it can receive recommendations, neighbors, friends and voluntary strangers. In addition, the reputation values are modified with time decay function, thus a bad host doesn't remain bad for life and a good host isn't considered good forever. Although model pays attention to aggregate reputation of groups, but weight assigned to each of them is predefined and agents are assumed to be benevolent and honest.

In [6] an adaptive technique has been proposed in e-commerce communities which is called peertrust. This framework identified the five factors, such as the total number of transactions of an agent, the credibility of the feedback given by an advisor, the transaction context and the community context factor for evaluating and quantifying the trustworthy of trustee. The model assumes that an agent with higher trust value always gives more trusty recommendation than an agent with a lower trust value, which is disadvantage in the case of the misleading behavior. Moreover, it doesn't consider the necessity of integrating direct trust and reputation into a unique value.

Also, another approach [10] was proposed to work in an open multi agent system. Specifically, the author incorporates direct trust, witness information (it takes into account attestations about the behavior of a trustee), role-based trust (it depends on the agent's relationships) and certified reputation (it's ratings presented by the rated agent about itself which have been obtained from its partners in past interactions [10]). There are several disadvantages of this model; these varieties of sources provide FIRE of a good versatility in many usual occurrences, but to correctly work a lot of parameters have to be tuned [4]. Also, in the Fire model agents are considered honest, in a similar way as [7].

The DHT trust model presented in [12] which uses full advantage of the Distributed Hash Table (DHT) to distinguish fake information from genuine of information. This system confects the distributed reputation into global reputation, but it doesn't verify direct trust of trustee. In addition, managers of groups cannot drive away the attacks such as collusion attacks.

The PBtrust model [18] was developed in service-oriented architecture (SOA) that considers different attributes and priorities. The PBTrust model derives the trustworthiness of a service provider from four perspectives: the provider's experience on the service, the similarity of priority distribution of attributes between the referenced service and the requested service, the suitability of the potential provider for the requested service and the time effectiveness of rating score from third parties. These features can give a more accurate to select service providers, but there is no mechanism to choose recommenders and they are suggested by servers that it becomes points of vulnerability for collusion attack. Moreover, credibility of recommendation does not discuss.

Against mentioned models with a fixed pre-determined ratio to combine direct trust and reputation, [4] is a trust model in which the integrating process is completely dynamic to obtain a synthetic measure. This value strictly depends on the number of interactions ($\beta 1$), recommendation reliabilities ($\beta 2$) and percentage of the agent of the community ($\beta 3$). The experimental results clarify that the usage of this model introduces significant advantages with respect to the usage of a static model [4]. Also, in [17] authors employed confidence factor, which is equivalent to the $\beta 1$ parameter used in [4], if agent i has had sufficient transactions with agent j, then i knows j well enough and

does not need many feedbacks from other agents about the trustworthiness of j. Otherwise, agent i will weigh more on the recommendations coming from other agents. However, as in most other trust models, updating trust to trustee and advisors process is assumed to have a static coefficient and time function does not have evaluation.

Summing up, few of trust approaches don't consider the reputation [9], but in the recent trust scenarios, like in the proposed model, a trustor behind its own direct trust, exploits also global reputation by considering the recommendation from other agents who have interacted with trustee. But there would be also some deficiencies when the trust of trustee is computed due to the presence of malicious advisors. Most of the existing relevant work tune weight of recommendation by using pre-defined value [6,7,11], exploiting discrepancies between computed trust and observed behavior [12] or even ignoring it [4,10]. Therefore, to avoid reputation miscalculation, in this paper a robust and efficient trust mechanism is proposed that filters this value and the reputation rates are suitably weighted. So normal distribution factor is used in order to uniform weight of recommendations to emphasize that advice, which is in a specific interval is more important as compared to other advice. Moreover, the client to operate its decision should combine some measures, such as direct trust and reputation or current observe and saved trust values to compute total trust that some models integrate two measures based on fixed ratio [7, 10].

In the DTM model, a common ontology is used to determine the dynamic weight of direct trust and reputation as in [4] and time function as in [7,10]. In a community consisting of competitive agents these factors are necessary but not sufficient. Meanwhile, the mentioned approaches generally do not consider the preference of the current or past experience [6,9,10,11,12] or integrating process is applied based on a static measure[4,7]. Consequently, we construct a dynamic framework to address this bottleneck and improve calculation accuracy. A series of experimental evaluations demonstrate validity of the proposed approach.

2. The proposed dynamic trust and reputation model proposal

2.1 General descriptions

In this section, Dynamic Trust Model (DTM) is proposed for e-commerce environments. Considering the drawbacks exposed in the previous section, a novel methodology has presented that addressing such issues that inhibits problems within a region where all participates are self-interested. The DTM model supposes the existence of a set of clients and servers that all agents can show a fraudulent behavior. Meanwhile, a client requests service to the server of an e-commerce system and can either be fully satisfied with the service or not and allocates trust data in range [0,1]. This measure is an evaluation of the quality of service (QOS) generally associated with the service provided by the server. But before requesting, when a client wants to interact with the server, it will first assess the trustworthiness of that to understand the expertise of the server. In order to perform such a trust decision a total trust score will be computed, whilst for each server taking into account two separate sources of information, namely; direct trust and reputation. Then total trust becomes the result of aggregation of them. **Figure 1** illustrates the flow of events in the DTM model.

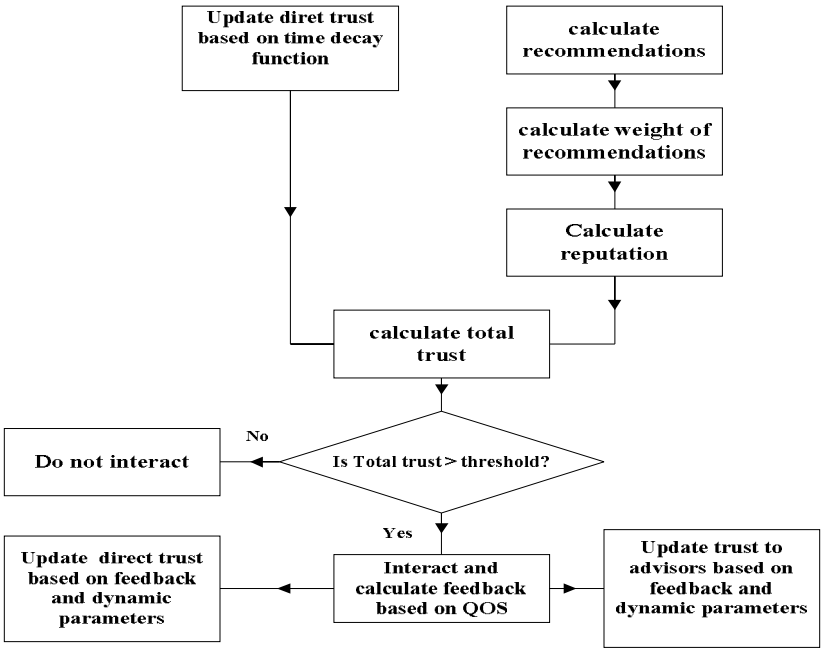

Figure 1: Flowchart of DTM Trust Model

2.2. Computing trust scores

Keeping in mind the discussion presented in the previous section, a framework is presented that provides a methodology for computing the overall trust measures involved in the mentioned scenario. Consider the situation where a client i wants to send a request to server j in order for server provider accomplishes a certain task. As stated before, each client icomputes a trust value for each candidate server j which is calculated from direct trust and reputation value. To this proposition, let T_{ij}denote the total trust value of server j from the viewpoint of client i; $TS_{ij}$and Rep_{ij}present the direct trust and reputation of server j, respectively. Trust measure is varied in [0,1] span that 0 value determines untrustworthy and one indicates that the agent is absolutely trustworthy.

In other hand, For a new agent, model knows nothing about it, hence Trust values of an unknown agent initialize by a minimum trust value and they can advance up the maximum of one during their activity on the community since researchers find that if a newcomer agent starts with pessimistic trust value, it reaches better results that are close to reality. The detailed methodology is divided into eight phases and is elaborated ahead as showed in **figure 2**.

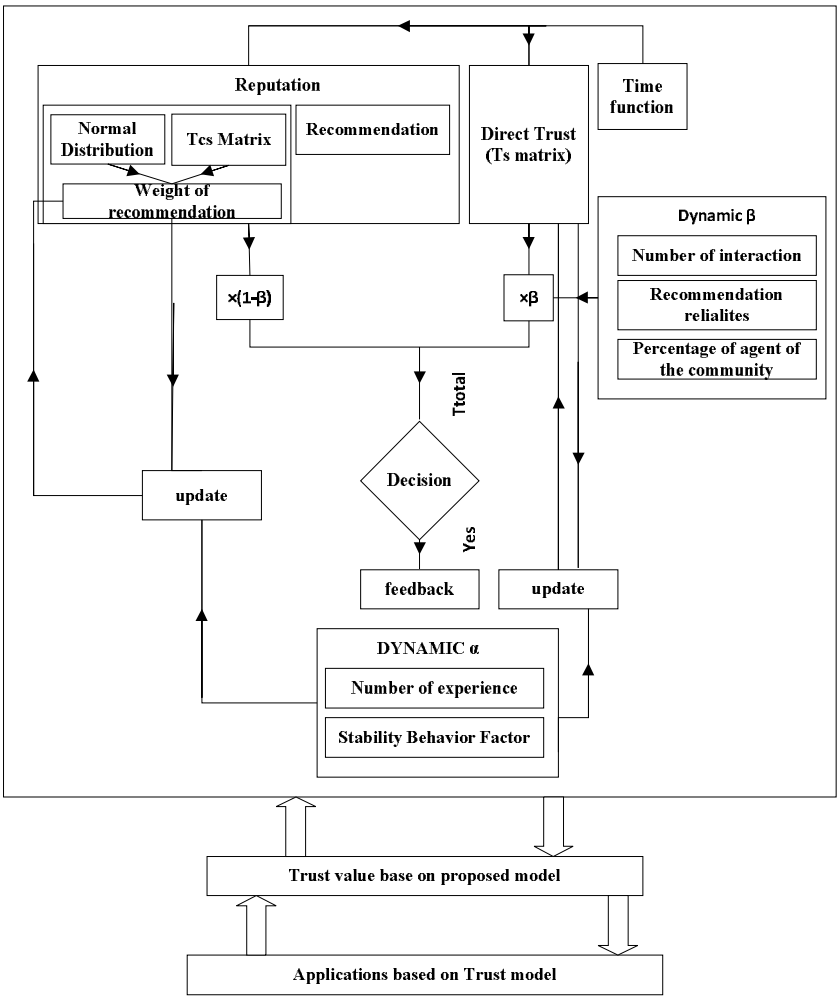

Figure 2: The structure of proposed Trust model

Phase 1: The first step is updating trustor's own information. The idea is to promote recent information and to deal with out-of-date information with less emphasis [16]. This way, useless knowledge has less impact on the final making judgment and can effectively reduce the impact of strategic attack (see section 4). So client i updates this item by (1) as follows.

$$TS_{ij} = n + \left(TS_{ij} - n\right) e^{-(t-t0)/\tau} \qquad (1)$$

Where,$TS_{ij}$is the trust valueat t= t_0, t_0 is the last time client i computed the trust of server j, n means the neutral value explained above. Besides the τ refers to an empirical constant that specifies how quickly or slowly trust information becomes invalid. More in particular, τ can be chosen based on how trustworthy a server is. If the TS_{ij}represents suspicious behavior, it will choose a bigger τ such that the $TS_{ij}$is less affected by time.

Phase 2: As said before, client i measures reputation value as one of the essential requirements of the system by soliciting opinion and then aggregating all the recommendations from other clients (Some trust models encourage the cooperation among advisors who are always ready to share their opinions with other clients, e.g. [4, 6, 14,15]). In addition, the credibility is factored into calculating the weight the recommender's feedback from client i's perspective.

At first, equation(2) updates TCS_{ijk} that is credibility's client k ($1 \leq k \leq n$) to server j from the viewpoint of client i. It adopted to improve precision and is similar to the time function used in the phase 1.

$$TCS_{ijk} = n + \left(TCS_{ijk} - n\right) e^{-(t-t0)/\tau} \qquad (2)$$

Typically, if client i maintains a general credibility value for each advisor, vicious client may provide a good rating for selected servers to maximize their credibility and subsequently provide unfair advice to defame the reputation of the other well-behaved server. Instead, in the DTM model every trustor has separate matrix for each advisor-server pair where each value $TCS_i(k,j)$ represents the credibility that client i assigns to the gossip coming from client k about server j. Additionally, it is reasonably accuracy of reputation value is affected by the credibility of recommendations. In other words, trust models must accurately filter out collected opinions from diverse testimony of different trustworthiness. But we believe the problem of unfair rating exists in almost mentioned models. So in this article, an approach is suggested that if the advisor gives an opinion too diverging from the expectation value of the server j, its credibility value is decreased and its report will have less impact on computing the reputation or even is discarded. To achieve this, normal distribution can be effectively applied as depicted in **figure 3**.

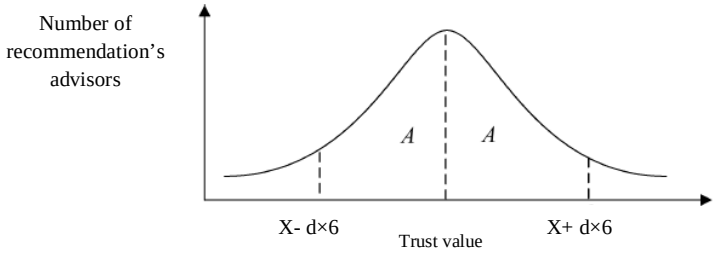

Figure 3. Distribution of Recommendation of clients

In (3), consider X is an average of client's recommendations and let 6^2 denote the variance of the Normal distribution that can be defined by (4); Where M_{kj} is client k's feedback about server j.

$$X = \frac{1}{n}\left(\sum_{k=1}^{n} M_{kj}\right) \qquad (3)$$

$$\sigma^2 = \frac{1}{n}\left(\sum_{k=1}^{n} M_{kj} - X^2\right) \qquad (4)$$

In this study's approach, there is a normal distribution factor that specifies how credible the recommendation or the reputation is. This concept determines the acceptable deviation and divides recommendations into two intervals: one of that is in interval A: [$X - d \times 6$, $X + d \times 6$], and the other is outside it, where d decides the altering size of the interval A as for different situation. This means that, there is a trade-off that the trustor will have to make based on the importance of the interaction. If the client k's rating is classified interval A, its credibility value will update and go up by (5). Otherwise, it is outside the span, a penalty will be applied and its credibility value will be decreased. In fact, a lower importance level is given to the recommendation by (5):

$$TCS_{ijk} = \begin{cases} TCS_{ijk} + e^{-|M_{kj} - X|} \times \left(1 - TCS_{ijk}\right) & , M_{kj} \in A \\ TCS_{ijk} - \left(1 - e^{-|M_{kj} - X|}\right) \times \left(1 - TCS_{ijk}\right) & , M_{kj} \notin A \end{cases} \qquad (5)$$

Generally, collusive agents assign each other a high trust value to artificially increase their reputation while bad mouthing servers outside its group. Additionally, some malicious clients build a good credibility and then abuse it to mislead other clients and defame trustee, also called strategic attack. Nonetheless, in fact, the difference between the recommendation's malicious agent and the other will be significant. Therefore, the normal distribution approach can be used as a false feedback filter to identify dishonest feedback from an honest one and will make the trust model robust against these attacks.

Phase 3: the reputation value $Rep_{ij}$of sever j from view point of i can be computed from (6) as the average of the recommendation received by other clients:

$$\mathrm{Re}\, p_{ij} = \sum_{k=1}^{n} \left(TCS_{ijk} \times M_{kj} \right) \Big/ N_{rep} \qquad (6)$$

Where N_{rep}represents the cardinality of the clients who provided a recommendation about server j.

Phase 4: as stated before, the trustori eventually computes the total trust by taking into account both direct trust and reputation that is multiplied by β,ϵ[0,1] as the corresponding weighting factor as follows:

$$T_{ij} = TS_{ij} \times (\beta) + Rep_{ij} \times (1-\beta) \quad (7)$$

Let T_{ij}denote the total trust value from client i to server j and β describes how confident client i is regarding its direct trust value on server j. Generally speaking a larger value of β gives heavier weight to the direct trust that as mentioned before, the same dynamic β is used as in [4].

Phase 5: After determining the total trust value, it is compared with thresholds of absolute trust and absolute mistrust which are either dependent of risk and criticality of iteration. When the total trust value is less than the absolute mistrust, the service won't be provided because server j appears untrustworthy and nothing happens. If the total trust falling in the probabilistic region between two thresholds, the final decision is affected by paranoid or trusting client i is. Otherwise, the total trust is higher than the threshold of absolute trust, client i shall be able to make a decision with an acceptable level of security and will proceed the iteration phase in order to accomplish the specified task.

Phase 6: When a transaction is done, client i can observe the Qos of received service.

Thus, the feedback ϵ [0,1] which reflects the trustworthiness of the agents involved in the last iteration, will be recorded as part of direct experience to update interval trust score. Updating procedure is illustrated in (8) and (9) where TS'_{ij} and $TCS'_{ikj}$are updated values of server j's direct trust and client k's credibility from the view point of client i, respectively:

$$TS'_{ij} = TS_{ij} \times (1-\alpha_s) + f_{ij} \times (\alpha_s) \qquad (8)$$

$$TCS'_{ikj} = TCS_{ikj} \times (1-\alpha_c) + \alpha_c (1 - |f_{ij} - M_{kj}|) \qquad (9)$$

In which, $f_{ij}$is the current feedback of server j, TS_{ij}andTCS_{ikj}refer the trust measures in the previous step, α_s and α_c are constants in the range between 0 and 1, represent the importance that client i gives to the past experience with respect to the current feedback. Generally speaking as α increase, it also gives more weight to the feedback of the interaction that just took place and less weight to previous trust values. Where α=0 indicates that the client i toward update the trust values in server j and other clients does not assign any importance to the current interaction and only previous values is considered. Vice versa, if α=1 the updating process only is impressed upon feedback without using the contribution of the past experience.

In the DTM trust model, the coefficient α is dynamically computed, which is depend on fluctuating behavior and number of iteration factors. Firstly, (10) computes α_s as the product of two contributions calledα_{s1} and α_{s2}.

$$\alpha_s = \alpha_{s1} \times \xi + \alpha_{s2} \times (1-\xi) \qquad (10)$$

Let α_{s1} denote fluctuating behavior and α_{s2} be the number of iteration factor, ξis in the range [0,1] which is chosen by the experimenter. $\alpha_{s1}$is defined in (11):

$$\alpha_{s1} = \left| f_{ij} - TS_{ij} \right| \qquad (11)$$

The expression $| f_{ij} - TS_{ij} |$ returns the fluctuating of current quality of service provided by j actually through past behaviors, in fact α_{s1} is sensitive to the variation of behavior. If the quality is rated either as previous experience and $| f_{ij} - TS_{ij} |=0$, this means that server j has completely routine behavior . Therefore, according to observed fixity behavior, current iteration will not be considered in order to update process.

If $| f_{ij} - TS_{ij} |=1$, this indicates that the server j's behavior is temperamentally due to improvement or various malicious motives. It is our belief that this differentiation is crucial. It is reasonable to assume that the iteration with

higher fluctuating should be weighted more than those with lower fluctuating. So α_{s1}=1gives complete preference to the current feedback which has maximum impact on α_s .α_{s1}parameterissupposed to linearly decrease from 1 to 0 that denote the absolute deviation and absolute stability, respectively.Note both increment and decrement in quality may result into this deviation. therefore, sensitivity to high α_{s1} values and giving maximum importance to disparate feedback provides an incentive for quality improvement and a debilitation mechanism of strategic attack by giving more weight to this iteration.

Afterward, the number of iteration factor is calculated as shown below in (12):

$$\alpha_{s2} = 1 - (1 - \mathrm{MIN}) * (x_{ij} / \mathrm{MAX}) \quad (12)$$

Where $x_{ij}$is a mapping that denotes the number of the last iteration between client i and server j, MAX represents a suitable number of iterations (whose value can be predetermined) and MIN is the minimum value that client i decides to assign to α_{s2} . The contribution α_{s2} is equal to1 if x_{ij}=0 (since this indicates that no iteration has been done and then it is possible to use only current trust value), while α_{s2} has its minimum value MIN when $x_{ij}$is equal to MAX, meaning that the importance of the previous Trust value has to be considered maximum when client-server pair has sufficient iterations and number of current iteration is equal to MAX. As explained in phase 2, malicious servers may camouflage themselves as honest ones by providing high quality services strategically in early iterations. To handle strategic disinformation α_{s2} will allocate lower weightage to small x_{ij}values which indicate initial past iterations and will concentrate on current feedback. Based on the above considerations, the following formula computes α_c in (13) which is obtained from (14) and (15):

$$\alpha_c = \xi\alpha_{c1} + (1-\xi)\,\alpha_{c2} \quad (13)$$

$$\alpha_{c1} = \left| \left(1 - \left(f_{ij} - M_{kj} \right) \right) - TCS_{ikj} \right| \quad (14)$$

$$\alpha_{c2} = 1 - (1 - \mathrm{MIN}) * (x_{ikj} / \mathrm{MAX}) \quad (15)$$

In equation (14), the expression | f_{ij} - M_{kj} | describes the difference between current feedback and client k's opinion, obviously the expression (1- (f_{ij}- M_{kj})) determines whether an advisor is honest or not. Similarly, the fluctuating behavior factor of advisors is estimated by comparing current honestly advice and previous trust value as shown in (14). Withal $x_{ikj}$is the number of last client k's rating to server j from the view point of client i. Note the exactly the same concepts are used as indicated in (10), (11) and (12).

3. Experimental results

In this section, the result of numerical experiments will describe in order to evaluate aspects of the proposed trust model, DTM, and demonstrate both its effectiveness and reliability. Also, they assess the accuracy of the DTM model and compare it with [4] scheme to show its advantages of being more robust again various attacks. After these, models and also attacks scenarios have simulated using Matlab code. Every experiment is started through the certainty of client and server provider. In simulation, we assume that there are 100 clients that are looking for a service provider to interact with and gain the provided service amongst 20,50 or 100 servers that are supposed to the provide services. For simplicity, we also assume that every server in the system provides the same kind of service with a different QOS. According to the real environment, agents are classified into honest agents and malicious agents in the experimental system. Honest agent provides expected quality of service and honest feedbacks and malicious agents include collusive agents and strategic agents (providing fake information to undermine the system performance).

Under these conditions, each agent is set well or wicked randomly, in fact a random sequence of service qualities $TS_{real\ 1}$ …$TS_{real\ NS}$ (where NS indicates the number of servers) and a sequence of credibilities$TCS_{real\ 1}$ ….$TCS_{real\ NC}$ (where NC denotes the number of clients) are created. These real values are as a hidden source of agent's behavior.

θ is defined as the percentage of malicious agents varied by increments of 10%, from 10% to 90%. For instance, θ=0. 5 means 50 percentages of community have vicious behavior, in the other hand $TS_{real}$and TCS_{real}measures less than 0.5 is considered malicious. Gained utility is used as a measurement for the quality of obtained service that depends on the performance of the server.

The simulation consists of a number of experiments in which the clients are run and enhance their direct knowledge about the environment to gain more utility as possible. In words, clients try to compute T_{ij}close to TS_{real} value. The other constants of tests are shown in **Table 1**.

Table1. Experimental Parameters

Parameter	Value	Description
d	0.05	It alters the size of the interval A
n	0.2	Newcomer's initial trust value
θ	0.5	Percentage of malicious agents
f	10	Maximum number of iterations that malicious agents may camouflage themselves
τ	0.2	Decay parameter
δ	0.2	Weight ratio current and past trust value
NS	20, 50 or 100	Number of servers
NC	100	Number of clients
Iteration	2000	Number of simulation rounds

3.1. Experiment 1: Successful transaction rate

As seen in fig 4, Successful transaction rate (SRT) of the DTM scheme and [4] model is computed under collusion and strategic attack. The metrics, Successful transaction rate is the ratio of the number of successful transactions over the total transaction number in the community up to a certain time. It is commonly used to judge the productivity and level of security of a trust model. It is obvious that a community with an effective trust mechanism should have a high Successful transaction rate as trustors are able to make an informed trust decision.

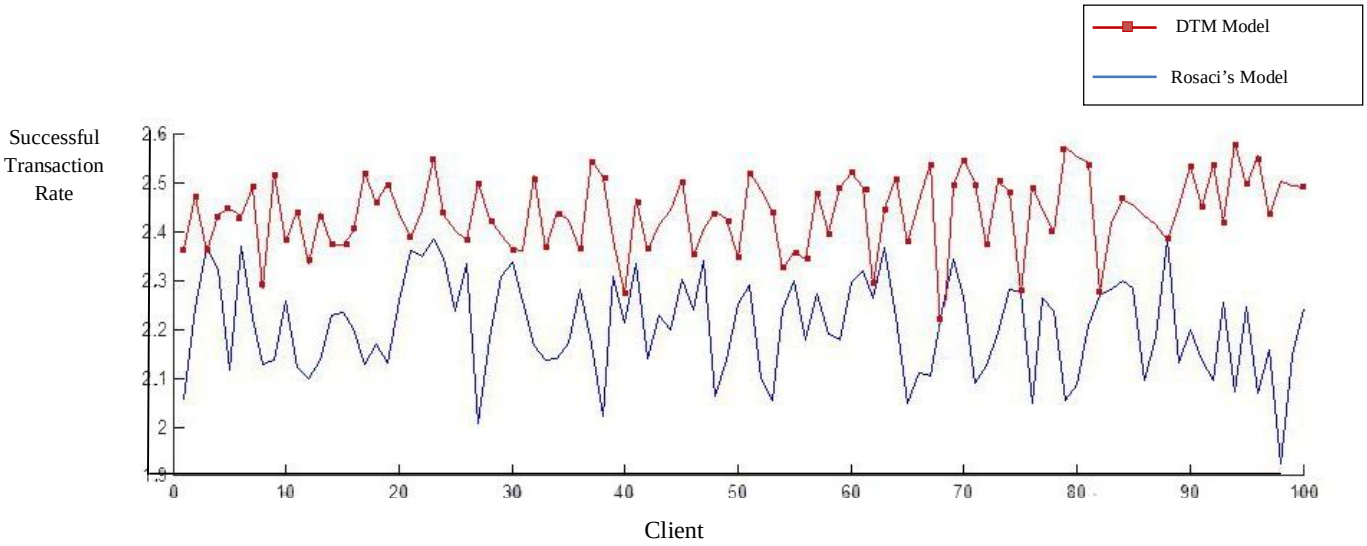

Figure 4. Comparison of DTM with Rosaci's Model [4] in terms of a successful transaction rates

The experiment starts by repeatedly having randomly selected client-server pair (NS=20 and NC=100) initiating iteration. Pairs then perform the transaction. So transaction succeeds are recorded and the Successful transaction rate is computed when the experiment proceeds. In **figure 4**, the dotted line represents the curve of the DTM that is higher than the curve of Rosaci's Model and it always stays at the top. In DTM, advisors with higher credibilities value have higher effectiveness by constructing a normal distribution filtering mechanism. Moreover, good agents are encouraged and malicious agents are deterred from participating in a process by fluctuating behavior factor. Additionally, compared to Rosaci's Model, time decay function and number of iteration factor are designed to tackle strategic attack. This experiment demonstrates that compared with Rosaci's Model, the DTM trust model improves the accuracy.

3.2. Experiment 2: DAC error and θ

Simulation consists of 50 servers and 100 clients among which some are trustworthy and some are untrustworthy. DAC error is used (Difference between Actual and Computed trust value) to measure the robustness of the trust model respect to percentage of malicious agent increasing in step of 10% (θ= 0.1,…0.9). It is reasonable, as the percentage of malicious agents increases, the chances for trustor to receive unfair information increase. Therefore, a lower DAC error indicates higher accuracy that is defined as follows:

$$DAC = \left(\sum_{j=1}^{NS} TS_{realj} - TS_{ij} \right) \Big/ NS \tag{16}$$

As argued before, $TS_{ij}$is the measured total trust score of the server that refers to Qos provided by server j and TS_{real}refers to the real value of server j's reliability. **Figure 5** shows the difference between the two trust models from the viewpoint of a client.

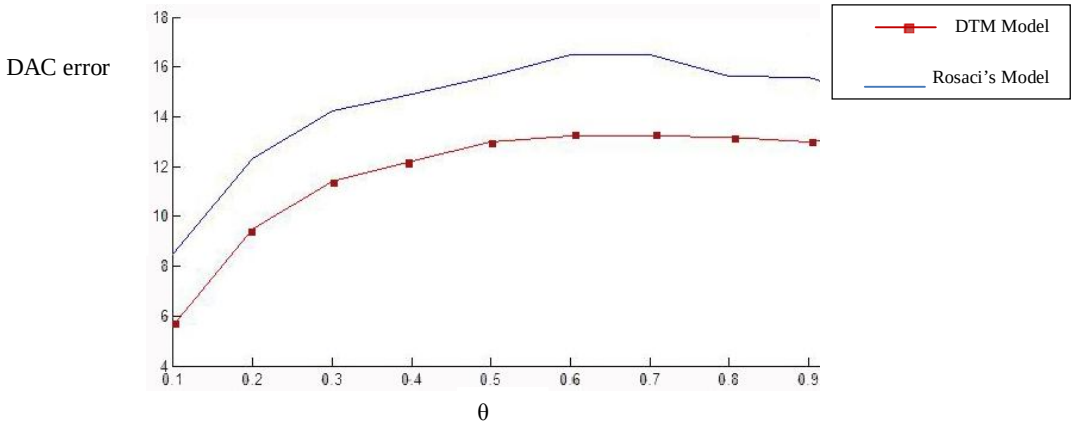

Figure 5: Comparison of DTM with Rosaci's Model [4] in terms of DAC error gained under θ increment

As seen above both trust models show an error increase when the percentage of malicious drives increase, nevertheless DTM model gains always lower DAC error than Rosaci's model. In general, agents with a more adaptive trust framework would be much more stable and thus obtained lower DAC error from the environment. So the DTM model is able to cope with collusive activities due to filter of recommendations and it introduces dynamic α coefficient to restrain strategic misbehavior. Therefore, DTM model is effective even in an overwhelming increment of the faction of malicious agents. However, Rosaci's Model doesn't provide any mechanism to control of changes in agents behavior and credibility computation that affects the accuracy of trust estimation in a very biased environment.

3.3. Experiment 3: DAC error and n

In the third experiment, we show the result of the increment of n factor in the DAC error. We have considered 10 different values of n, namely n=0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1 which represent possible newcomer's initial trust values that denote paranoid to trust views. For each of these values, we have run 2000 iterations; the result of an experiment is reported in **figure 6**.

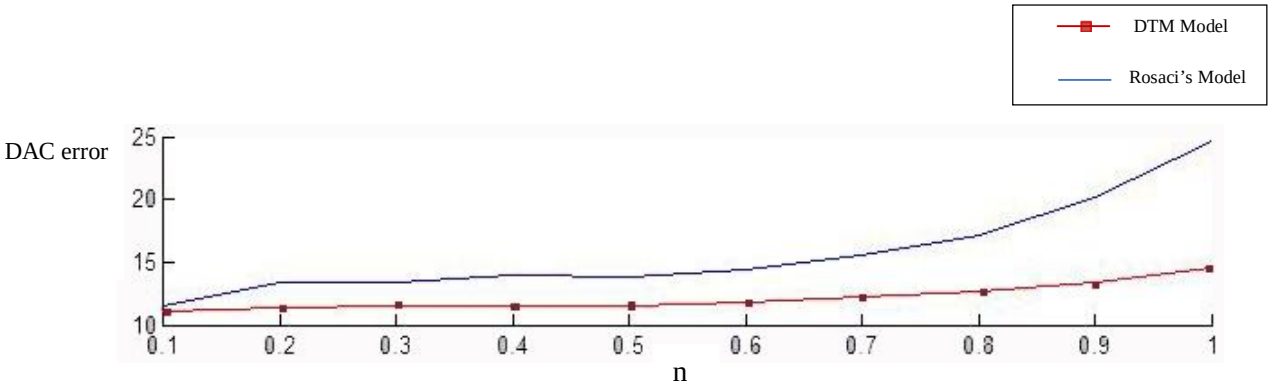

Figure 6: Comparison of DTM with Rosaci's Model [4] in terms of DAC error gained under an increment

As seen in **figure 6**, the two models perform almost equally well when the n is set to absolute paranoid. The increment of the initial trust value makes trustor more optimistic and consequently implies an increment of DAC error. Withal, like the previous experiment, in all 10 cases DTM error diagram stays lower than Rosaci's model whilst the DAC error of Rosaci's model deteriorates. These observations can be explained as follows. As n grows, trustor assigns higher trust values to any malicious agents, so they are able to more easily subvert the trust model. In this situation, Rosaci's model relies on high credibility of a bad advisor while in DTM the credibility of malicious advisors is extremely dropping by comparison with all recommendations; it overrides the effect of the false information. Moreover, DTM uses α_{2c} and α_{2s} factors to offset the risk of optimistic initial trust since notwithstanding high TS and TCS value, the first f iterations do not have a significant impact on total trust and agents need more time to prove itself to the trustor.

4. Conclusion

Trust and reputation approaches can be used to predict trustworthiness of an anonymous server and isolate malicious agents in a self-organized and large-scale e-commerce community. In this paper, a novel trust model is presented called DTM to enhance the accuracy in choosing a trustworthy server to interact with, in the presence of malicious agents. Recently it has been argued some research works in this field, but they are most susceptible to certain attacks. Therefore, building on previous trust models, the trust assessment procedure is designed to trade off the damage of unfair information by the authentic information in a more efficient manner. In fact, DTM mostly focused on issues related to implementation security and resistance to malicious attacks, such as collusion and strategic behavior by introducing a variety of features that are quantified through dynamic parameters. The simulation results confirm the correctness of DTM model and its ability to efficiently react against behavioral fluctuations and dynamic conditions. As future work, we intend to study other different aspects of the environment to find what improvements can be made to improve the robustness of DTM to fight against more intelligent attacks. We also plan to apply the current proposal between parties in security protocols such as SSL/TLS and voting protocols to make them more secure.

REFRENCES

[1] S. M.Furnell, et al., 2008, Securing Information and Communication Systems Principels Technologies and Application. Boston London: Artech House.

[2] W. El-Hajj, January 2012, The most recent ssl security attacks origins implementation evaluation and suggested countermeasures, Security and Communication Networks, vol. 5, pp. 113–124.

[3] A. Mathur and S. KrSharma, 2011, Sniffing: A major Threat to Secure Socket Layer and its Detection, International Journal of Computer Application(IJCA), pp. 135-139.

[4] D. Rosaci, April 2012, Trust measures for competitive agents, International Journal of Knowledge-Based Systems, vol. 28, pp. 38–46.

[5] D. Gambetta, 2000, Can We Trust Trust?,Trust: Making and Breaking Cooperative Relations, Basil Blackwell, Oxford:Basil Blackwell chapter 13, pp. 213-237.

[6] L. Xiong and L. Liu, June 2003, A Reputation-Based Trust Model for Peer-to-Peer ecommerce Communities, presented at the Proceedings of the 4th ACM conference on Electronic commerce USA, pp.228-229.

[7] G. Derbas, et al., July 2004, TRUMMAR - a trust model for mobile agent systems based on reputation, presented at the Proceedings of the The IEEE/ACS International Conference on Pervasive Services, USA, pp. 113 -120

[8] lsaacPinyol and J. Sabater-Mir, July 2011, Computational trust and reputation models for open multi-agent systems: a review, Artificial Intelligence Review-AIR,Springer Netherlands,Volume 40, Issue 1 , pp. 1-25.

[9] S. P. Marsh, April 1994, Formalising Trust as a Computational Concept, PHD thesis, Department of Computing Science and Mathematics University of Stirling,UK.

[10] T. D. Huynh, et al., March 2006, An integrated trust and reputation model for open multi-agent systems, Autonomous Agents and Multi-Agent Systems, vol. 13, pp. 119–154.

[11] S. D. Kamvar, et al., 2003, The EigenTrust Algorithm for Reputation Management in P2P Networks, presented at the 12th International conference on World Wide Web Conference, New York, USA, pp. 640-651.

[12] Y. Liu, et al., July 2012, DHTrust: A Robust and Distributed Reputation System for Trusted Peer-to-Peer Networks, Concurrency and Computation: Practice & Experience, vol. 24, pp. 1037–1051.

[13] L. X. Chen Jia, XiaocongGan, Wenhui Liu, and Zhangang Han, January 2012, A Trust and Reputation Model Considering Overall Peer Consulting Distribution, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans archive, vol. 42, pp. 164-177.

[14] N. K. S. Vibha Gaur, PunamBedi, November2011, A Dynamic Model for Sharing Reputation of Sellers among Buyers for Enhancing Trust in Agent Mediated e-market, International Journal of Computer Science Issues (IJCSI), vol. 8, pp. 143-153.

[15] C.-L. Cheng, et al., February 2012 , METrust: A Mutual Evaluation-base Trust Model for P2p Network, international journal of Automation and Computing, vol. 9, pp. 63-71.

[16] B. Khosravifar, et al., June, 2012, CRM: An efficient trust and reputation model for agent computing, Knowledge-Based Systems archive, vol. 30, pp. 1–16.

[17] C. T. B. Yangc, October 2011, R2Trust, a reputation and risk based trust management framework for large-scale, fully decentralized overlay networks, Future Generation Computer Systems archive, vol. 27, pp. 1135–1141.

[18] M. Z. a. Xing Su a, Yi Mua, QuanBai, August 2013, A robust trust model for service-oriented systems, Journal of Computer and System Sciences, vol. 79, pp. 596–608.

[19] Y. C. Zheng Yana, YueShenb, August 2013, A practical reputation system for pervasive social chatting, Journal of Computer and System Sciences, vol. 79, pp. 556–572.

[20] G. n. P. r. Fe´ lix Go´mezMa´rmol, May 2012, TRIP, a trust and reputation infrastructure-based proposal for vehicular ad hoc networks, Journal of Network and Computer Applications, vol. 35, pp. 934–941.

Authors

M. Asadzadeh Kaljahi received the B.S. degree in Computer Software from the Islamic Azad University of parand, Iran, in September 2009 and the M.S. degree in information technology-information security from the University of Tehran, Iran in September 2012. She is researching interest is in peer-to-peer trust and reputation systems and security protocols.

Ali. Payandeh received B.Sc. and M.Sc. degrees in electrical engineering from TarbiatModarres University, Iran, in 1991 and 1994, respectively, and the Ph.D. degree in electrical engineering from K. N. Toosi University of Technology, Iran, in 2006. From 1991 to 1995, he was a faculty member in the Department of Electrical Engineering at Malek-Ashtar University of Technology, Iran. Since 1996, he has been a Director of Research at the Applied Science Research Association (ASRA), Iran, where he has been involved in research in secure satellite communications. His research interests include information theory, coding theory, secure communications and satellite communications

M.B. Ghaznavi-Ghoushchi received the B.Sc. degree from the Shiraz University, Shiraz, Iran, in 1993, the M.Sc. and Ph.D. degrees both from the TarbiatModares University, Tehran, Iran, in 1997, and 2003 respectively. During 2003-2004, he was a researcher in the TMU Institute of Information Technology. He is currently an Assistant Professor with Shahed University, Tehran, Iran. His interests include VLSI Design, Low Efficient circuit and systems, Computer Aided Design Automation for Mixed Signal and UML-based designs for SOC and Mixed InformationLow-Power and Energy-Mixedor Mixed-Signal.