A Robust and Scalable Reputation System for Secured Communication in Multi Agent Systems

G.Gomathi[1] and N.Radhika[2]

[1]*Department of Computer Science and Engineering, PRIST University - Trichy Campus, Trichy, India.*
Email ID: gomsgopal@gmail.com

[2]*Department of Computer Science and Engineering, PRIST University - Trichy Campus, Trichy , India.*
Email ID: vnradmca@yahoo.com

Abstract —Security and privacy issues have become critically important with the fast expansion of multi agent systems. Most network applications such as pervasive computing, grid computing, and P2P networks can be viewed as multi agent systems which are open, anonymous, and dynamic in nature. Such characteristics of multi agent systems introduce vulnerabilities and threats to providing secured communication. One feasible way to minimize the threats is to evaluate the trust and reputation of the interacting agents. Many trust/reputation models have done so, but they fail to properly evaluate trust when malicious agents start to behave in an unpredictable way. Moreover, these models are ineffective in providing quick response to a malicious agent's oscillating behavior. Another aspect of multi agent systems which is becoming critical for sustaining good service quality is the even distribution of workload among service providing agents. Most trust/reputation models have not yet addressed this issue. So, to cope with the strategically altering behavior of malicious agents and to distribute workload as evenly as possible among service providers; we present in this paper a dynamic trust computation model called "Secured Trust." In this paper, we first analyze the different factors related to evaluating the trust of an agent and then propose a comprehensive quantitative model for measuring such trust. We also propose a novel load-balancing algorithm based on the different factors defined in our model. Simulation results indicate that our model compared to other existing models can effectively cope with strategic behavioral change of malicious agents and at the same time efficiently distribute workload among the service providing agents under stable condition

***Keywords* — Multi agent system, trust management, reputation model, loads balancing, malicious behaviour.**

1. INTRODUCTION

In a multi agent system (MAS), agents interact with each other to achieve a definite goal that they cannot achieve alone and such systems include P2P grid computing, the Semantic We, pervasive computing, and MANETs. Multi agent Systems are increasingly becoming popular in carrying valuable and secured data over the network. Nevertheless, the open and dynamic nature of MAS has made it a challenge for researchers to operate MAS in a secured environment for information transaction. Malicious agents are always seeking ways of exploiting any existing weakness in the network. This is where trust and reputation play a critical role in ensuring effective interactions among the participating agents. Researchers have long been utilizing trust theory from social network to construct trust models for effectively suppressing malicious behaviors of participating agents. Trust issues have become more and more popular since traditional network security approaches such as the use of fire wall, access control, and authorized certification cannot predict agent behavior from a "trust" viewpoint.
A routing algorithm determines the sequence of channels for a packet to traverse from the source to the destination. Several routing algorithms were proposed for meshes and tori.

However, global reputation models are much more complex to manage than local experience models as malicious agents have the opportunity to provide false feedbacks. With these research problems in mind, we propose a feedback-based dynamic trust computation model named Secured Trust which can effectively detect sudden strategic alteration in malicious behavior with the additional feature of balancing workload among service providers. Secured- Trust considers variety of factors in determining the trust of an agent such as satisfaction, similarity, feedback credibility, recent trust, historical trust, sudden deviation of trust, and decay of trust. We have used a novel policy of utilizing exponential averaging function to reduce storage overhead in computing the trust of agents. We have also proposed a new load-balancing algorithm based on approximate calculation of workload present at different service providers.

2. RELATED WORK

Bayesian network-based trust model believes that trust is multidimensional and agents need to evaluate trust from different aspects of an agent’s capability. This model uses Bayesian network and Bayesian probability to calculate trust. This model’s main flaw lies in the authors’ assumption that all the agents have identical Bayesian network architecture which is unrealistic because different agents have different requirements which leads to different network architecture. In the case of aggregating recommendation from other agents, this model assumes that all the agents are truthful in providing their feedbacks. The network is highly dynamic and unpredictable, trust values should decay as time elapses in absence of interaction. However, these models fail to simulate real life decay function which has a small decay rate in the initial phase while displaying higher decay rate as more and more time elapses. We have incorporated such decay function in our trust model along with many other issues which have not been addressed by existing trust models. Another aspect which is slowly becoming vital for sustaining service quality is the balanced distribution of workload among service providers. Almost all trust models have ignored this issue so far. In fact, none of the models discussed so far address the aspect of balancing load among the trusted agents for proper maintenance of service quality. In a trust computing environment, an agent with the highest trust is normally selected as service provider, so highly reputed agents handle bulk of the total servicer quests. We have proposed such a load balancing algorithm.

3. SECUREDTRUST

3.1 Satisfaction

Satisfaction function measures the degree of satisfaction an agent has about a given service provider. In other words, it keeps record of the satisfaction level of all the transactions an agent makes with another agent. However, instead of storing all of the transaction history, we have defined an exponential averaging update function to store the value of satisfaction. This greatly reduces the storage overhead and at the same time assigns time relative weight to the transactions. Let Sat (p; q) represent the amount of satisfaction agent p has upon agent q based on its service up to n transactions in the time interval.

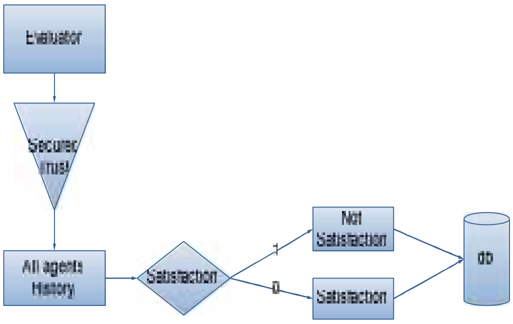

Figure 1 Satisfaction Trust

3.2 Feedback Creditability

Feedback credibility is used to measure the degree of accuracy of the feedback information that the recommending agent provides to the evaluator. Normally, it is assumed that good agents always provide true feedback and malicious agents provide false feedback. However, this is not always the real scenario as good agents might provide false feedbacks to their competitors and malicious agents might occasionally provide true feedbacks to hide their real nature. So, feedback credibility is needed to determine the reliability of the feedback. During trust evaluation, feedbacks provided by agents with higher credibility are trust worthier, and are therefore weighted more than those from agents with lower credibility.

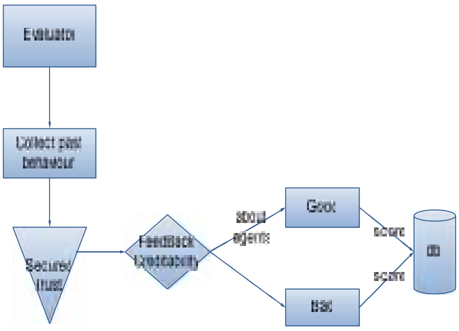

Figure 2.Feedback creditability

3.3Trust computation

Direct trust:

Direct trust also known as local trust represents the portion of trust that an agent computes from its own experience about the target agent. Let DT (p; q) represent the direct trust that agent p has upon agent q up to n transactions in the t th time interval.

Indirect trust:

Indirect trust also referred as recommendation is computed from the experience of other agents. An agent utilizes the experience gained by other agents in the system to make effective transaction decisions especially when it has no or very little experience with the given target agent. To do so, an agent requests other agents to provide recommendation about the target agent. The evaluating agent then aggregates recommendation from other agents along with the Feedback credibility of the recommenders. Let IT(p; q) represent the indirect trust that agent p computes about agent q.

Recent trust:

Recent trust reflects only the recent behaviors. We have defined recent trust as a weighted combination of direct and indirect trust. Direct trust is given higher weight as the evaluating agent performs more and more interactions with the target agent, i.e., the evaluator becomes more confident about its own experience than taking recommendation from others. Let RT (p; q) represent the recent trust that agent p has upon agent q.

Historical trust:

Historical trust is built from past experience and it reflects long-term behavioral pattern. With the elapse of time, recent trend becomes historical trend, and as a result, we have defined historical trust by using an exponential averaging update function. By using an exponential averaging update function, we not only reduce the storage overhead associated with storing the previous recent trusts but also assign time relative weights to all the previous values. Let HT (p; q) represent the historical trust that agent p has about agent q.

Expected trust:

Expected trust reflects expected performance of the target agent and it is deduced from both recent and historical trust. In other words, we are combining both recent trend and historical trend to get a prediction of the future trend. Let ET (p; q) represent the expected trust of agent q from agent p's perspective.

Expected trust reflects expected performance of the target agent and it is deduced from both recent and historical trust. In other words, we are combining both recent trend and historical trend to get a prediction of the future trend. Let ET (p; q) represent the expected trust of agent q from agent p's perspective.

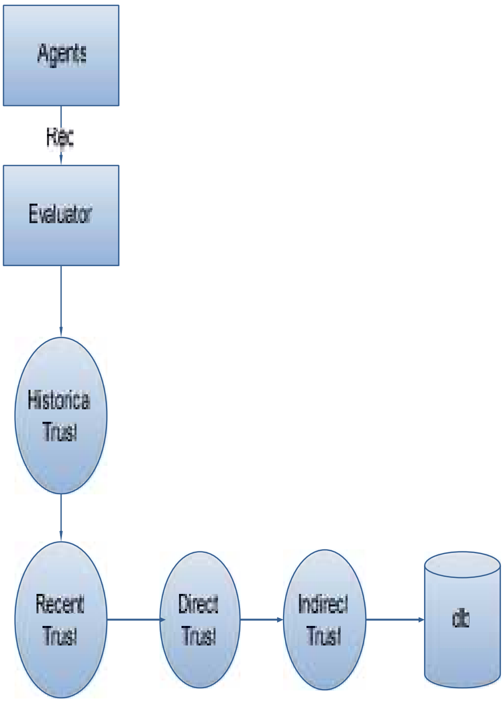

Figure 3.Trust computation

4. LOAD BALANCING AMONG AGENTS

We propose an algorithm for balancing loads among the trusted agents. For selective scenario, we first compute the trust of agents who respond to a transaction request and then we select the agent with the highest trust value. However, in this scenario, the agent with the highest trust value will have immense workload while other capable agents with slightly lower reputation will have considerably less workload. From this disproportionate allocation of workload is that the quality of service will fall greatly due to the heavy workload present at the highly trusted agents. So, a load-balancing algorithm is required for sustaining good service quality. In our load-balancing algorithm, we either calculate a heuristic value of workload and choose the agent with the smallest load or make a probabilistic choice based on the computed trust value of agents.

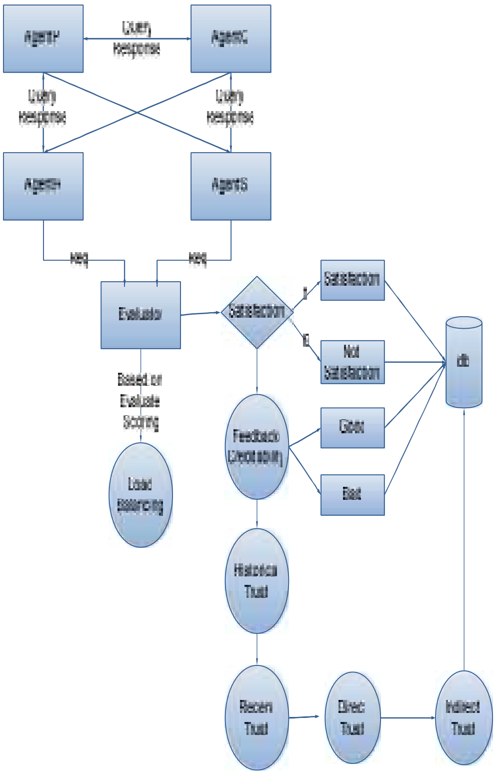

Figure 4. Load balacing agents

5 EXPERIMENTAL EVALUATIONS

This section evaluates Secured Trust's performance and shows its effectiveness under different adversarial strategy. We have carried out our experiment to achieve four main objectives. First, we evaluate its accuracy in terms of trust computation in the presence of malicious agents under two settings. The second experiment shows how quickly it adapts to strategically oscillating behavior. In the third set of experiments, we demonstrate the robustness of Secured-Trust compared to other existing trust models under different scenarios. Lastly, we show its effectiveness under the load-balancing scheme.

5.1 Agent's Behavioural Pattern

The behavioural pattern of good agents is quite easy to simulate as they provide good service and honest feedback .However, it is challenging to simulate an agent's malicious behaviour realistically. We mainly study three behavioural patterns, namely—non collusive, collusive, and strategically altering. In no collusive setting, malicious agents cheat during transaction and give false feedback to other agents, i.e., they rate good agents poorly while rating malicious agents highly. The collusive setting is similar to then on collusive setting with one additional feature that malicious agents form a collusive group and deterministically help each other by performing numerous fake transactions to boost their own rating while disparaging other good agents. We have used the parameter collusion to denote the Percentage of malicious agents forming a collusive group. In the strategically altering setting, a malicious agent may occasionally decide to cooperate in order to confuse the system. We use the parameter malicious res to model the rate of dishonest feedback by a malicious agent. In this case, other agents are commonly fooled into thinking that the Malicious agent is actually a good agent.

5.2 Performance Evaluation Index

To compare the performance of Secured Trust with other existing trust models, we use a evaluation index named, successful transaction rate (STR). STR is described as the ratio of the number of successful transactions to the total number of transactions. Since computed trust values may range differently for different trust models, other form of evaluation index such as trust computation error is not suitable for comparison. It really does not matter what range of trust value we assign to an agent, what matters is how efficiently the model can filter out malicious agents based on the calculated trust value. In other words, the relative ranking of agents based on their trust values is comparable and that's why we only compute STR for comparison with other models. We determine STR against the variation of malicious_per, malicious_res, and collusion. All experimental results are averaged over 30 runs.

6 CONCLUSIONS

We have presented a novel trust computation model called Secured Trust for evaluating agents in multivalent environments. Secured Trust can ensure secured communication among agents by effectively detecting strategic behaviours of malicious agents. In this paper, we have given a comprehensive mathematical definition of the different factors related to computing trust. We also provide a model for combining all these factors to evaluate trust and finally, we propose a heuristic load-balancing algorithm for distributing workload among service providers. Simulation results indicate, compared to other existing trust models, Secured- Trust is more robust and effective against attacks from opportunistic malicious agents while capable of balancing load among service providers

REFERENCES

[1]. N.R. Jennings, "An Agent-Based Approach for Building Complex Software Systems," Comm. ACM, vol. 44, no. 4, pp. 35-41, 2001.

[2]. R. Steinmetz and K. Wehrle, Peer-to-Peer Systems and Applications. Springer-Verlag, 2005.

[3]. Gnutella, http://www.gnutella.com, 2000.

[4]. Kazak, http://www.kazaa.com, 2011.

[5]. Edonkey2000, http://www.emule-project.net, 2000.

[6]. I. Foster, C. Kesselman, and S. Tucker, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," Int'l J. High Performance. Computing Applications, vol. 15, no. 3, pp. 200-222, 2001.

[7]. T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web,"Scientific Am., pp. 35-43, May 2001.

[8]. D. Saha and A. Mukherjee, "Pervasive Computing: A Paradigmfor the 21st Century," Computer, vol. 36, no. 3, pp. 25-31, Mar. 2003.

[9]. S.D. Ramchurn, D. Huynh, and N.R. Jennings, "Trust in Multi-Agent Systems," The Knowledge Eng. Rev., vol. 19, no. 1, pp. 1-25, 2004.

[10]. P. Dasgupta, "Trust as a Commodity," Trust: Making and Breaking Cooperative Relations, vol. 4, pp. 49-72, 2000.

[11]. P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman,"Reputation Systems," Comm. ACM, vol. 43, no. 12, pp. 45-48, 2000.

[12]. A.A. Selcuk, E. Uzun, and M.R. Pariente, "A Reputation-Based Trust Management System for P2P Networks," Proc. IEEE Int'lSymp. Cluster Computing and the Grid (CCGRID '04), pp. 251-258, 2004.

[13]. M. Gupta, P. Judge, and M. Ammar, "A Reputation System forPeer-to-Peer Networks," Proc. 13th Int'l Workshop Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'03), pp. 144-152, 2003.

[14]. K. Aberer and Z. Despotovic, "Managing Trust in a Peer-2-PeerInformation System," Proc. 10th Int'l Conf. Information and Knowledge Management (CIKM '01), pp. 310-317, 2001.

[15]. L. Mui, M. Mohtashemi, and A. Halberstadt, "A ComputationalModel of Trust and Reputation for E-Businesses," Proc. 35th Ann.Hawaii Int'l Conf. System Sciences (HICSS '02), pp. 2431-2439, 2002.

[16]. L. Mui, "Computational Models of Trust and Reputation: Agents,Evolutionary Games, and Social Networks," PhD thesis, MIT,

http://groups.csail.mit.edu/medg/medg/people/lmui/docs, Dec. 2002.

[17]. F. Cornelli, E. Damiani, S.D. Capitani, S. Paraboschi, and P.Samarati, "Choosing Reputable Servents in a P2P Network," Proc. 11th ACM World Wide Web Conf. (WWW '02), pp. 376-386, May2002.

[18]. Y. Wang and J. Vassileva, "Bayesian Network-Based TrustModel," Proc.

IEEE/WIC Int'l Conf. Web Intelligence (WI '03), pp. 372-378, Oct. 2003.

[19]. S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," Proc. 12th ACM Int'l World Wide Web Conf. (WWW '03), pp. 640-651, 2003

[20]. L. Xiong and L. Li, "Peertrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 7, pp. 843-857, July 2004.

[21]. M. Srivatsa, L. Xiong, and L. Liu, "TrustGuard: Countering Vulnerabilities in Reputation Management for Decentralized Overlay Networks," Proc. 14th ACM Int'l Conf. World Wide Web(WWW '05), pp. 422-431,2005.

[22]. Z. Runfang and H. Kai, "PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing," IEEE Trans. Parallel and Distributed Systems, vol. 18, no. 4, pp. 460-473, Apr. 2007.

[23]. B. Li, M. Xing, J. Zhu, and T. Che, "A Dynamic Trust Model for the Multi-Agent Systems," Proc. IEEE Int'l Symp. Information Processing (ISIP '08), pp. 500-504, 2008.

[24]. Y. Zhang, S. Chen, and G. Yang, "SFTrust: A Double Trust Metric Based Trust Model in Unstructured P2P Systems," Proc. IEEE Int'l Symp. Parallel and Distributed Processing (ISPDP '09), pp. 1-7, 2009.

[25]. [25] J. Hu, Q. Wu, and B. Zhou, "FCTrust: A Robust and Efficient Feedback Credibility-Based Distributed P2P Trust Model," Proc. IEEE Ninth Int'l Conf. Young Computer Scientists (ICYCS '08),pp. 1963-1968, 2008.

[26]. X. Wang and L. Wang, "P2P Recommendation Trust Model," Proc. IEEE Eighth Int'l Conf. Intelligent Systems Design and Applications(ISDA '08), pp. 591-595, 2008.

[27]. L. Wen, P. Lingdi, L. Kuijin, and C. Xiaoping, "Trust Model of Users' Behavior in Trustworthy Internet," Proc. IEEE WASE Int'l Conf. Information Eng. (ICIE '09), pp. 403-406, 2009.

[28]. R. Aringhieri, E. Damiani, S.D. Capitani, S. Paraboschi, and P.Samarati, "Fuzzy Techniques for Trust and Reputation Management in Anonymous Peer-to-Peer Systems: Special Topic Sectionon Soft Approaches to Information Retrieval and InformationAccess on the Web," J. Am. Soc. for Information Science andTechnology, vol. 57, pp. 528-537, 2006.

[29]. E. Damiani, S.D. Capitani, S. Paraboschi, and P. Samarati,"Managing and Sharing Servants' Reputations in P2P Systems,"IEEE Trans. Knowledge and Data Eng., vol. 15, no. 4, pp. 840-854,July/Aug. 2003.

[30]. D. Wen, W. Huaimin, J. Yan, and Z. Peng, "A Recommendation-Based Peer-to-Peer Trust Model," J. Software, vol. 15, no. 4,pp. 571-583, 2004.

[31]. J. Sabater and C. Sierra, "REGRET: A Reputation Model for Gregarious Societies," Proc. Fourth Workshop Deception, Fraud and Trust in Agent Societies, pp. 61-69, 2001.

[32]. J. Sabater and C. Sierra, "Social Regret, A Reputation Model Based on Social Relations," ACM SIGecom Exchanges - Chains of Commitment, vol. 3, pp. 44-56, Dec. 2001.

[33]. J. Sabater and C. Sierra, "Reputation and Social Network Analysis in Multi-Agent Systems," Proc. First Int'l Joint Conf. Autonomous Agents and Multi-Agent Systems (AAMAS '02),pp. 475-482, 2002.

[34]. L. Xiong and L. Liu, "A Reputation-Based Trust Model for Peer-to-Peer Ecommerce Communities," Proc. Fourth ACM Conf. Electronic Commerce (EC '03), pp. 228-229, 2003.

[35]. T.D. Huynh, N.R. Jennings, and N.R. Shadbolt, "An Integrated Trust and Reputation Model for Open Multi-Agent Systems,"Autonomous Agents

and Multi-Agent Systems, vol. 13, no. 2, pp. 119- 154, 2006.

[36]. N.R. Jennings, T.D. Huynh, and N.R. Shadbolt, “FIRE: AnIntegrated Trust and Reputation Model for Open Multi-Agent Systems,” Proc. 16th European Conf. Artificial Intelligence (ECAI ’04), pp. 18-22, 2004.