Volume 5, Issue 7, July 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

# Enhanced EigenTrust Model for Peer-To-Peer Networks

**Preeti Sondhi*, Anuradha Panjeta**
CSE & Kurushetra University,
Haryana, India

***Abstract—"Evolving Cloud Computing has released vast opportunities to share large amount of distributed resources globally and conveniently with significant pace. Despite the potential to provide low cost security, its ever increasing popularity and usage has put forward new challenges related to security and trust for the users as well as for the service providers. Selfish behaviour of nodes in the wireless sensor network further complicates the task. This paper proposes an improvised Eigen Trust algorithm along with enhanced normalization and remunerates trust transitivity clause, an algorithm which reduces the number of inauthentic files spread in the network, by isolating malicious peers from it. Finally the proposed mechanism has been tested under a simulated environment and the results have been presented."***

***Keywords— Cloud Computing; Security Issues; Trust; Reputation Systems; Wireless Sensor Network; Peer to Peer Network, EigenTrust Model, Enhanced EigenTrust Model (EET).***

## I. INTRODUCTION

Cloud Computing has resulted from the convergence of Grid Computing. Cloud computing is an innovative Information System (IS) architecture, visualized as what may be the future of computing, a driving force demanding from its audience to rethink their understanding of operating systems, client–server architectures, and browsers. Cloud computing has leveraged users from hardware requirements [1], while reducing overall client side requirements and complexity. As cloud computing is achieving increased popularity, concerns are being voiced about the security issues introduced through the adoption of this new model.

Peer-to-Peer networks is a fast developing branch of Computer Science and many researchers are developing new algorithms for such systems. After the success of systems like Gnapster, Kazza, the use of such networks became a necessity. Peer-to-Peer (P2P) networks represent an environment well suited for distributed reputation management. In P2P networks, every node plays the role of both client and server, and is therefore sometimes called a servent [2]. There are two phases in the use of P2P networks. The first is a search phase, also called service discovery, which locates servents offering the enquired service. The second phase is the actual service usage, for instance the download of a certain information item. In some P2P networks, the service discovery relies on information stored in centralised service repositories. One such example is Napster with its resource directory server. In pure P2P networks however, like Gnutella and Freenet, also the service discovery phase is completely distributed without single centralised components. There are also hybrid architectures, e.g. the FastTrack architecture which is used in P2P networks like KaZaA, grokster and iMesh. In FastTrack based P2P networks, there are nodes and supernodes. The latter keep track of other nodes and supernodes that are logged onto the network at any one time, and thus act as directory servers during the search phase. The download phase normally consists of requesting the service from one of the nodes that was discovered during the search phase. Unfortunately the open and anonymous nature of these systems, makes the surveillance of the network almost impossible, leaving many open doors for malicious individuals, who want to profit from what the system is offering.P2P networks introduce a range of security threats, as they can be used to spread malicious software, such as viruses and Trojan horses, and easily bypass firewalls. There is also evidence that P2P networks suffer from free riding [3]. In this paper we will focus on the Trust Management for the P2P networks trying to explore the possibilities of this kind of management in completely decentralized systems. We will describe the Enhance EigenTrust Algorithm proposed in for this problem.

Trust Management is a mechanism that allows to establish mutual trust (allows participants to a network to cooperate in a game-theoretic situation that corresponds to the repeated prisoner dilemma and leads in the long term to an increase aggregated utility for the participants". The best framework for explaining the notion of Trust Management is the e-commerce framework and the most representative example is eBay online auction system. In eBay, users are selling and buying products one from each other and accumulate experience with each transaction made. When a user wants to buy a product, he will search for users who are selling that product. He will decide to buy from a seller with a good experience. After the transaction is done both parts will rate one each other depending on how the transaction finished and the actual rating will be add to the previous experience. The same happens when a user wants to sell a product. Thus in this kind of

 

systems users are encouraged to behave nice because otherwise they will be isolated by all other users and thus from the system. In eBay the experience or reputation of an user, represents the trust that all other participants puts in that user. It is worth noticing that this reputation score is unique per user, so is a global trust score. We will follow this idea of user ratings after each transaction in the development of the algorithm. The main problem with eBay Trust Management is that it is a completely centralized system where the reputation score is managed by some servers (eBay authority).

Reputation systems are well suited to mitigate against these problems, e.g. by exchanging and sharing information about rogue, unreliable or selfish participants. Many authors have proposed reputation systems for P2P networks [4], [5], [6], [7], [8], [9]. The purpose of a reputation system in P2P networks is:

1) To determine which servents are most reliable at offering the best quality resources, and

2) To determine which servents provide the most reliable information with regard to 1.

The Eigen Trust algorithm proposed by Kamvar et al. (2003) [10] is aimed at deriving global reputation scores in P2P communities with the purpose of assisting members in choosing the most reputable peers. Eigen Trust assumes that each peer i observes whether its interactions with a peer j have been positive or negative. The satisfaction score $s_{ij}$ for peer j as seen by peer i is based on the number of satisfactory interactions $sat(i,j)$ and the number of unsatisfactory interactions $unsat(i,j)$, and is expressed as:

$$= sat(i.j) - unsat(i.j) \quad (1)$$

The normalised local trust score $c_{ij}$ of peer j as seen by peer i is computed as:

$$c_{ij} = \frac{\max(s_{ij},0)}{\sum_{l \in L} \max(s_{i,l},0)} \quad (2)$$

where L is the local set of peers with which peer i has had direct experiences. This step effectively normalises the local trust values to the range [0,1] and thereby removes any negative trust values. A local peer with a large negative satisfaction score would thus have the same normalised local trust score as a local peer with satisfaction score 0.

## II. LITERATURE REVIEW

### *A. A Trust Computing Mechanism for Cloud Computing*

Mohamed Firdhous, Osman Ghazali, Suhaidi Hassan [11]: In this paper, the authors propose a trust formulation and evolution mechanism that can be used to measure the performance of cloud systems. The trust system provides a trust score between 0 and 1 for different levels of services and continues to improve these values based on the performance of the system. Hence the proposed system would be more useful for providing differentiated services at different quality levels. The proposed mechanism has been evaluated using a simulation environment setup with Octave the open source Matlab clone. The simulation results show that the proposed system works satisfactorily under constrained simulated environment. The authors propose to carry out this in a future research.

### *B. Cloud Computing System Based on Trusted Computing Platform*

Zhidong Shen, Li Li, Fei Yan, Xiaoping Wu [13]: In this author analyze some security requirements in cloud computing environment. Since the security problems both in software and hardware, they provided a method to build a trusted computing environment for cloud computing by integrating the trusted computing platform (TCP) into cloud computing system. They Propose a new prototype system, in which cloud computing is combined with Trusted Platform Support Service (TSS) and TSS is based on Trusted Platform Module (TPM). In this design, better effect can be obtained in authentication, role based access and data protection in cloud computing environment. Trusted cloud computing is built on the trusted computing platform and can provide flexible security services for users.

## III. PROBLEM STATEMENT

For EigenTrust Model, a peer in one situation can achieve trust value 1 by doing 1 successful download. Another peer can achieve trust value 1 after 10000 downloads in another situation. It is unfair and may lead to Sybil attacks. Although $c_{ij} = c_{ik}$ , peer k has more trustable performance than peer j. **Using Equation (1) and (2)**

$$c_{ij} = \frac{s_{ij}}{\sum_j s_{ij}}, \qquad s_{ij} = sat(i.j) - unsat(i.j)$$

$$c_{ij} = \frac{1}{1} = 1, c_{ik} = \frac{10000}{10000} = 1$$

Case of Trust Transitivity: In case trust values are not normalized, trust value of indirect peer interactions will be significantly higher than trust values of direct peer interactions since the transitive trust is computed over multiplication of trust values for all the peers involved in the path involved.

For an analogy, consider a node A has trust score of 2 for node B and similarly B has trust score of 4 for node C. Assumingly, node A has no direct interaction with node C, thus trust value of node C for node A would be 2*4 which is illogical since that would result into a system design wherein indirect interactions will always be encouraged.

## IV. PROPOSED SOLUTION

### *A. EigenTrust Algorithm*

In EigenTrust, the global reputation of each peer i is given by the local trust values assigned to peer i by other peers, weighted by the global reputations of the assigning peers.

 

**Basic Eigen Trust Algorithm**

$$\vec{t}^{(0)} = \vec{p}$$

**repeat**

$$\vec{t}^{(k+1)} = (1-a)C^T \vec{t}^{(k)} + a\vec{p}$$

$$\delta = \left\| \vec{t}^{(k+1)} - \vec{t}^{k} \right\|$$

until $\delta < \epsilon$

Like in eBay, in a P2P file-sharing system peers will rate each other after every transaction made. If peer i downloaded a file from peer j then if the file was authentic it will set $tr(i,j) = 1$. otherwise it sets $tr(i,j) = -1$. We set the local trust to $s_{ij} = \sum tr(i,j)$. More formally if:

$sat(i,j)$ = the number of satisfactory transaction

$unsat(i,j)$ = the number of unsatisfactory transaction

***Remark 1***: The local trust value that peer i have in peer j is:

$$s_{ij} = sat(i,j) - unsat(i,j)$$

The main problem with this values is that we cannot aggregate them. One reason will be that the values of $s_{ij}$ cannot be well interpreted. For example the fact that a peer i has $s_{ij}$ = 10 for a peer j and another peer k (different from i and j) has $s_{kl}$ = 1000 would not tell us too much, because it is possible that peer i is new and have made only 10 transaction with peer j, and peer k has made lots of transactions: sat(k,l) = 2000 and unsat(k, l) = 1000. Thus aggregating these values would not help. Another drawback is that malicious peers can report a very high local trust, subverting the system. To overcame this we will normalize the local trust in the following manner:

***Remark 2:*** The new normalized local trust, $c_{ij}$ that peer i has about peer j is

$$C_{ij} = (s_{ij} - Min_i)\frac{Avg_i}{2(Max_i - Min_i)}$$

Wherein, $Min_i = \forall_j Min(s_{ij})$, $Max_i = \forall_j Max(s_{ij})$and $Avg_i = \frac{\sum_j s_{ij}}{Number\ of\ Direct\ Interactions}$ , But for Min = Max, $C_{ij} = (s_{ij})\frac{Avg}{2}$.

Next step is to aggregate these normalized local trust values. From now on, we will use local trust values for normalized local trust values.

***1) Global Trust Values:*** One big problem with local trust values is that peer i doesn't know all peers in the network and usually only for few j the local trust is nonzero. Suppose now that peer i wants to download from peer k that we have never seen, thus $c_{ik} = 0$. A natural way for peer i to know about the trust of peer k is to ask all his friends (here friends means the peers with which peer i has interact) what they are knowing about peer k. But because not all friends are trusty it will be better to weight there opinion about peer k. This technique is called transitive trust. So the trust that peer i will place in peer k is defined by:

$$t_{ik} = \frac{\sum_j C_{ij}(C_{ij} + C_{jk})}{C_{jk}}$$

Then we can see that we can write $t_{ij}$ in matrix notation. If $C = (c_{ij})$ is the matrix of all local trust values, then we write $\overrightarrow{t_i} = C^T \overrightarrow{ci}$, where $\overrightarrow{t_i}$is the vector containing all values $t_{ik}$.

## V. RESULTS

In this research, TRMSim-WSN is used for simulation. All the experiments carried out consisted of 100 WSNs whose nodes were randomly distributed over an area of 100 square units. Of the nodes, requesting 100 times a certain service and applying a specific trust and/or reputation. Number of sensors used in the simulation is 50 and simulated for 100 executions. Another assumption in this simulation, every node only knows its neighbours within its RF range. Simulation parameters and default values used in the experiments are summarized in Table 1 below.

Table 1: Simulation Parameters

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| Number of executions | 100 | Malicious nodes (%) | Variable |
| Number of Networks | 100 | Plane (Units) | 100 |
| Minimum Number of Sensors | 50 | Delay between simulated networks | 0 |
| Maximum Number of Sensors | 50 | Radio range | 12 |
| Clients (%) | Variable | Security threats used | Collusion and Oscillating |

### *A. Comparison with Well-Known Trust and Reputation Models*

***1) Average Accuracy:*** The average accuracy factor is the important factor that indicates the security level. High average accuracy means that the model is secure. The results displayed in Table 5.2 and graphically represented in Fig 1, Fig 2, and Fig 3 prove that EET(Enhanced Eigen Trust) is generally more secure than other methods

 

| Trust Models | WSN without Collusion/ oscillating | Collusion | Oscillating |
|---|---|---|---|
| EIGEN TRUST | 44% | 85% | 85% |
| PEER TRUST | 59% | 15% | 80% |
| POWER TRUST | 78% | 87% | 90% |
| BTRM-WSN | 60% | 39% | 90% |
| EET | 80% | 69% | 91% |

Table2: Comparison between EET and well-known trust and reputation models (malicious nodes percentage ≈ 60%)

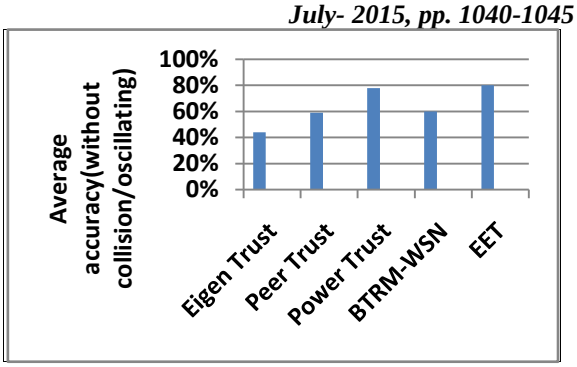


Fig 1: Comparison between EET model and existing trust and reputation models in terms of average accuracy with collision/oscillating

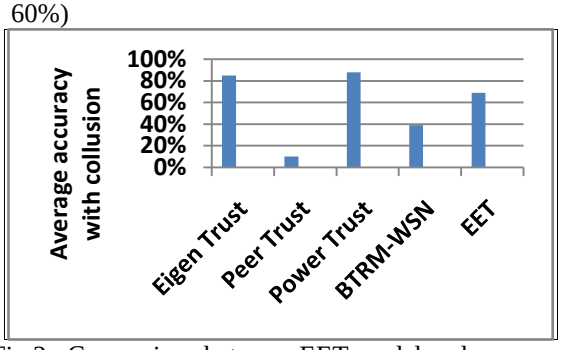


Fig 2 : Comparison between EET model and and reputation models in terms of average accuracy with collusion

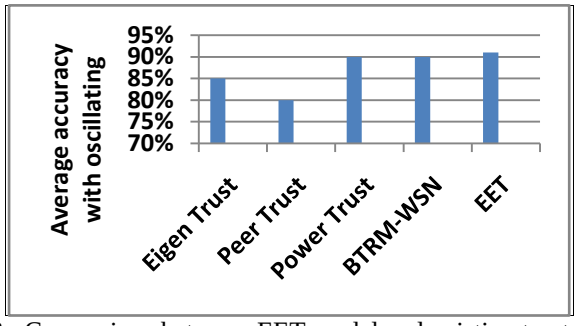


Fig 3 : Comparison between EET model and existing trust existing trust and reputation models in terms of average accuracy with oscillating

In summarizing the results, it can be seen that EET is resilient to Oscillating effects and its accuracy and scalability remain high while results are optimal to WSN with or without collusion or oscillating threats.

***2) Average Path Length:*** This factor indicates network efficiency and availability. Shorter average path length indicates that energy consumption is low and the network throughput is high due to increase in network lifetime. Results of this comparison are provided in Table 3

| Trust Models | WSN without Collusion/oscillating | Collusion | Oscillating |
|---|---|---|---|
| EIGEN TRUST | 7.5 | 7.4 | 6.4 |
| PEER TRUST | 7 | 6.8 | 6.5 |
| POWER TRUST | 6.5 | 7 | 7 |
| BTRM-WSN | 5.8 | 2.9 | 4.5 |
| EET | 4.67 | 2.71 | 3.96 |

Table 3: Comparison between EET and well-known trust andreputation models in terms of average path length (MaliciousNodes Percentage ≈ 60%)

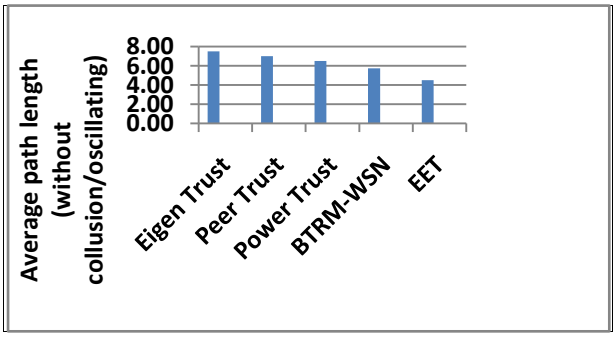


Fig 4: Comparison between EET model and existing trust and reputation models in terms of average accuracy without collision/ oscillating


Fig 5: Comparison between EET Model and existing trust and reputation models in term of average path length with collusion

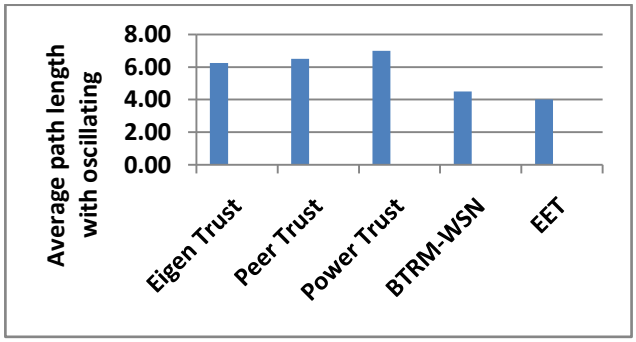


Fig 6: Comparison between EET Model and existing trust and reputation model in terms of average path length with oscillating

 

Results in Fig 4 show that EET has an average length less than other mechanisms during simulation for WSN without applying oscillating and collusion threats. Fig 5 and Fig 6 prove the quality of the EET model and that it is energy aware rather compared to other models during effects of oscillating and collusion threats. To summarize these results, EET has less average path length than other models during WSN simulation with or without threat tests which means that it performs packet transfer from source to destination with less energy consumption.

***3) Energy Consumption:*** Energy consumption is the main indicator of network lifetime. High energy consumption causes a network to die in a short time. Table 4 shows the energy consumption values for EET and other well-known models under collusion and oscillating effects.

| Trust Models | Collusion | Oscillating |
|---|---|---|
| PEER TRUST | 3.7 * 10^15.0 | 5.6 * 10^15.0 |
| POWER TRUST | 3.7 * 10^15.0 | 1.2 * 10^17.0 |
| BTRM-WSN | 5.8 * 10^15.0 | 5.3 * 10^17.0 |
| EET | 1.5 * 10^13.0 | 2.1 * 10^16.0 |

Table4: Comparison between EET and well known trust and reputation models in terms of Energy Consumption(mj)

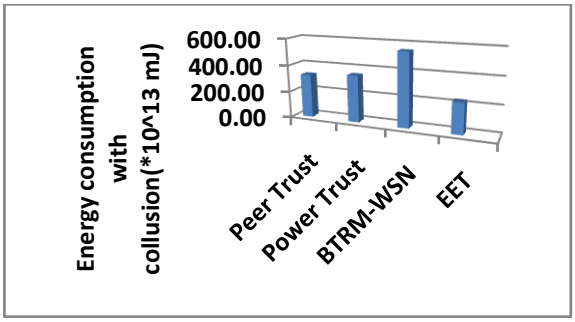


Fig 7: Comparison between EET and well known trust and reputation models in terms of average energy consumption with collusion

From Fig 7 it can be noted that energy consumption is very much lower than other models during collusion effects. In the networks under oscillating effects EET shows a decrease in energy consumption compared to PowerTrust and BTRMWSN

***B. Results Discussion***

Results indicate that EET rather than other well known models shows flexibility in strength and accuracy toward malicious nodes. Energy consumption by using EET is lower than using other models. Accordingly, it can be summarized that the contribution of this research is as follows:

- The EETmodel is resilient to collusion effects. Accuracy and scalability remain high for static WSNs and increase with increasing number of client sensors.
- Comparing well-known trust and reputation models such as EIGEN TRUST, PEER TRUST, PowerTrust, and BTRM-WSN with EET shows that EET has better average accuracy and less average path length than other mechanisms, due to the security and energy aware.

## VI. CONCLUSION

Selfish behaviour of nodes in the wireless sensor networks puts forward a challenging task in order to promote security/trustworthiness of the network. This paper proposes improvised EigenTrust algorithm along with enhanced normalization and redefining trust transitivity clause, an algorithm which reduces the number of inauthentic files spread in the network, by isolating malicious peers from it.

We have seen that we can compute global trust values in a distributed and secure way. The experiments weren't so bad, and we have seen that this method really works (at least in the simulation). Also the overhead imposed by the computation was acceptable. We don't say that this algorithm is perfect, but it can be the base of future research in the field of Trust Management for P2P systems.

Future work can be focused on creating new libraries of EET framework to extend support for the model. The major improvement of our proposed EET system compared to its two competitors is that it increases the accuracy in searching for trustworthy sensors, and thus provides a higher level of security. While future work will keep on developing the algorithms searching for trustworthy sensors to improve the easiness in finding trustworthy sensors as well as the energy efficiency of our approach

## REFERENCES

[1] Dimitrios Zissis, Dimitrios Lekkas, *Addressing cloud computing security issues*, Future Generation Computer System 28 (2012), 583-592.

[2] Audun Josang, Elizabeth Gra, and Michael Kinateder, *Simplification and Analysis of Transitive Trust Networks*, School of Software Engineering and Data Communications Queensland University of Technology, Australia, Trinity College Dublin, Ireland, Institute of Parallel and Distributed Systems Faculty of Computer Science, University of Stuttgart, Germany.

[3] E. Adar and B. Huberman, *Free Riding on Gnutella*, First Monday (Peer-reviewed Journal on the Internet), vol. 5, no. 10, p. 8, October 2000.

[4] K. Aberer and Z. Despotovic, Managing *trust in a peer-2-peer information system*, in Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM01), H. Paques, L. Liu, and D. Grossman, Eds. ACM Press, 2001, pp. 10.317.

 

[5] F. Cornelli et al., Choosing *Reputable Servents in a P2P Network*, in Proceedings of the eleventh international conference on World Wide Web (WWW'02). ACM, May 2002.

[6] E. Damiani et al., *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 (CCS'02). ACM, 2002, pp. 207.216.

[7] D. Fahrenholtz and W. Lamesdorf, *Transactional Security for a Distributed Reputation Management System*, in Proceedings of the Third International Conference on E-Commerce and Web Technologies (EC-Web), vol. LNCS 2455. Springer, September 2002, pp.214.223.

[8] M. Gupta, P. Judge, and M. Ammar, *A reputation system for peer-to-peer networks*, in Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video (NOSSDAV), 2003.

[9] C. Liau et al., *Efficient Distributed Reputation Scheme for Peer-to-Peer Systems*, in Proceedings of the 2nd International Human.Society@Internet Conference (HSI), vol. LNCS 2713. Springer, 2003, pp. 54.63.

[10] S. Kamvar, M. Schlosser, and H. Garcia-Molina, *The EigenTrust Algorithm for Reputation Management in P2P Networks*, in Proceedings of the Twelfth International World Wide Web Conference, Budapest, May 2003.

[11] Mohamed Firdhous, Osman Ghazali, Suhaidi Hassan, *A Trust Computing Mechanism For Cloud Computing*, InterNetWorks Research Group, Universiti Utara Malaysia, Sintok, Kedah Darul Aman, Malaysia 2011 ITU-T

[12] Chang, L, Ti ; Chin L; Chang, A.Y.; Chun J , *Information security issue of enterprises adopting the application of cloud computing*, IEEE 2010 Sixth International Conference on Networked Computing and Advanced Information Management (NCM),pp 645, 16-18 Aug. 2010.

[13] Zhidong Shen, Li Li, Fei Yan, Xiaoping Wu, "*Cloud Computing System Based on Trusted Computing Platform*, International Conference on Intelligent Computation Technology and Automation, 2010 IEEE DOI 10.1109/ICICTA.2010.724

 