A Novel Trust Management Scheme Using Fuzzy Logic for a Pervasive Environment

V. Rhymend Uthariaraj, J. Valarmathi, G. Arjun Kumar,
Praveen Subramanian, and R. Karthick

Ramanujan Computing Center, Anna University, Chennai, India
valar.sakthi@gmail.com

Abstract. One of the most critical issues in distributed system is security. The ideal solution to this concern is to have an environment that is fully trusted by all its entities. Our proposed approach is truly unique and fully comprehensive incorporating fuzzy logic for subjective concepts and integrating various trust related characteristics. The trust values(TV) calculated between the entities are stored in a global data store. This TV can be used as a basis for future transactions of the entity concerned. Since the global data store contains the trust value of all the users of the environment, these trust values are more accurate than that which has been calculated through recommendations. This also obviates the necessity for entities to know each other beforehand, which is a prerequisite in the case of recommendations. Also the security issues related with the global trust scheme is analyzed and an efficient solution to it using fuzzy logic has been proposed.

Keywords: Trust, Global Trust Model, Pervasive Environment, Fuzzy Logic.

1 Introduction

In pervasive environments the nodes are highly mobile. So we can not implement formal cryptographic methods of security. And also formal methods of security implementation are highly processor intensive which is not a desirable feature in pervasive devices which are characterized by their low computing power. So trustworthy computation of security could be used as it is less processor intensive and provides better solutions overcoming the problems in the formal method.

The trust value of an environment gives the quantitative measure of how much that environment is dependable. This trust can be calculated based on the interactions with the environment directly and also from the result of the interaction with other users. This information from other users have been so far calculated based on recommendations from these users directly. But this would result in the loss of some valuable information. So in our proposed scheme we are moving towards a global data store which contains the trust information obtained from cumulative user interactions about the various environments and also the users of this environments.

And also there is a constraint that we should have prior knowledge about atleast a few nodes in the new environment. This has a very low probability when we are in a new environment. Some of the Security issues which are associated with the global

N. Meghanathan et al. (Eds.): NeCoM, WiMoN, and WeST 2010, CCIS 90, pp. 144–152, 2010.

trust model should be found out and an efficient method to overcome these problems should be given.

The paper is organized as follows. Section 2 consists of related works in the areas of trust and pervasive computing, section 3 explains about the proposed work, section 4 shows our model implementation in a hospital scenario, section 5 discusses about the performance evaluation as compared with the deterministic model and finally section 6 consists about the conclusion and future works.

2 Related Works

The development of pervasive computing from isolated "smart spaces" to generic ubiquitous systems has heralded a new era in the development and sophistication of humans. We follow the definition of trust as in [2], that "Trust is the subjective probability by which an individual, A, expects that another individual, B, to perform a given action the way it wants on which its welfare depends". The importance of trust in pervasive computing is underscored by the low processor intensity and the unpredictability of the new devices that enter a pervasive environment. . This has led to trust being used as a basis for providing security and as as mentioned in [1], "a trustable entity is believed to to secure". In [3], trust was quantified to be between -1 to +1, which we have retained in our paper. The proposed negotiation scheme that was suggested in [6] is found to be wanting in situations where in the users of a systems cannot identify new entrants, in which case , a global trust storage and retrieval mechanism is found to be more useful. Also in [6], a global trust model in a peer to peer network was demonstrated that has the potential to be extended to a pervasive environment.

Trust is not a concept unique to pervasive computing; it has been widely used in many other disciplines. Work [4] builds a trust model to improve mobile ad hoc networks' collaborations based on a Bayesian network.

In paper [15] use of fuzzy logic in a wireless sensor network has been described. And also in paper [16] the various use of fuzzy logic in trust calculation of a pervasive environment has been described. In that paper use fuzzy logic in the decision making for access control has also been discussed.

3 Proposed Work

In the traditional trust value computation, the trust value is computed by using recommendation from neighbor nodes. In our method we are trying to store the calculated trust value in a global data store which can then be the basis for future transactions.

3.1 Global Data Store

The trust values will be stored in a global data store which contains information about all the environments and users, and they should be made available anywhere, anytime on request. This could be achieved by accessing these data through the various wireless communications available everywhere.

The data value of each tuple contains <T,N,E,t> where, T → Average trust value of the environment will be between -1 to +1, where -1 implies complete mistrust, 0 implies no trust and +1 implies complete trust, N → Number of previous users of the environment, E → Type of the environment, t → Time of update of trust value.

3.2 Local Data Store

The local data store is specific to each environment or user and it is used to store information about the recent users of that environment or entity. This data value can be used as a basis because an entity can be assumed to retain its behavior in the same environment. This data is stored in the local machine for the fast retrieval of trust value. It gives better result than global data store because each user-environment relationship will be distinct.

Each tuple in the local data store is represented as <T,E,t> where, T → Trust value of the environment, E → Type of environment, t → Time of update of data.

3.3 Credential Matching

When a new device is detected in the environment, the environment must try to establish communication with the device. For this the environment first transfers some of its credentials to the device. If the device finds the credential satisfying its requirements, then the device would in turn send its own credentials to the environment. The environment also checks them with its requirements. Then both the device and the environment uses the credentials obtained to look up the trust values of each other in the data store.

3.4 Trust Negotiation

The trust value of the environment or the user must be calculated for establishing connection with them. This trust value could be obtained from the local data store if it is available and from the global data store. The following algorithm would gives the details about calculation of the initial trust for a new user.

```
INPUT     :   New user enters the environment
OUTPUT    :   NONE
ALGORITHM :
  Step 1 : Get the trust value of the user ,T_O(U) and
           time t_O from database
  Step 2 : If information about the user is already
           available  with the environment get it as
           T_E(U) and the time of interaction t_E
  Step 3 : Get current time, t
  Step 4 : Set weighing factor W_O and W_H such that
           Weighing factor => W_O+W_E =1, where W_E < W_O ,if
           previous info about user is available
```

Weighing factor => W_O =1, if previous information about the user is not available

Step 5 : $T_H(U) = W_O * T_O(U) * e^{-(t-tO)} + W_E * T_E(U) * e^{-(t-tE)}$

Step 6 : Return $T_H(U)$

Step 7 : End

Once the trust value is computed it would be checked with the threshold value, which could be specified by the environment. If the trust value is greater than the threshold value then the connection is established.

3.5 Trust Computation

Trust computation can be done as in [1]. The initial trust value calculated should be updated based on the various actions by the user. This trust update should be done at specific intervals of time.

Let us consider two devices A and B, where the trust value of B computed by A is $T_A(B)$. This value can be calculated depending upon the outcomes(degree of success) of a set of dedicated set of actions that underline the transaction. Some of the possible policies that define the actions are response time, infrastructure offered by the environment, cost etc.

Let us consider $C_A(B)$ as the adjustment factor of the trust computation. The new value of trust is computed as follows

$$T_A(B) = \begin{cases} 1 & \text{if } T'_A(B) + C_A(B) \geq 1 \\ T'_A(B) + C_A(B) & \\ -1 & \text{if } T'_A(B) + C_A(B) \leq -1 \end{cases}$$

Where T'A(B) is the old trust value.

The adjustment value is calculated based on the result of the various outcomes of the interactions between the entities. The value of CA(B) is between [-1,1]. We calculate the adjustment factor as the sum of the positive effect and the negative effect of the various actions of B towards A.

$C_A(B)$ is the summation of the policy vector μ which depends upon the entity. It is given by

$$\mu=[t1_A(B)\ \ t2_A(B)\ \ t3_A(B)\ \ \ldots\ \ tn_A(B)]$$

Each entry in the vector represents the outcome of a policy based trust calculation after an interaction. $t_A(B)$ is calculated as follows:

$$t_A(B)=P_A(B)+N_A(B)$$

where $P_A(B)$ → Positive result of the interaction

$N_A(B)$ → Negative result of the interaction

3.6 Updating Trust Values

The trust value computed by the user must be updated in the global and local data store. The algorithm for updating the data store is provided below:

INPUT:Trust value given by user about environment, $T_U(E)$

Trust value of the user from the environment, $T_E(U)$

OUTPUT : NONE

ALGORITHM :

Step 1: Get the old trust value of the environment $T_O(E)$

Step 2 : Get the average trust value of the users of the environment, $T_{avg}(U)$

Step 3 : Get number of previous interactions in the environment, N

Step 4 : If $T_E(U) = -1$ goto step 9

Step 5 : $T_n(E)=[(N*T_O(E))+(T_U(E)*(T_E(U)+1)/T_{avg}(U)]/(N+1)$

Step 6 : $T_{newavg}(U)= [T_{avg}(U) + (T_E(U) + 1)]/(N+1)$

Step 7 : N=N+1

Step 8: Get current time, t

Step 8 : Update values $T_n(E)$,N,(env),t

Step 9 : End

The larger the trust value of the computing entity and the larger trust placed upon it by the entity for which trust is calculated, the greater the influence on the global data store. An untrusted entity cannot affect the trust value of another significantly in the data store.

4 Security Issues

4.1 Malicious Nodes

Malicious nodes are the ones which spreads wrong information about an entity and ultimately affects the overall trust value of the entity. So the damage caused by the malicious nodes needs to be minimized. This is already taken care of by our formula (2) and (3).

Before two entities begin to interact both of them will give permission to the DBA to allow the entities that could modify their trust values. Without the permission of the entity the DBA will not allow any other entity to modify the trust value of that entity.

4.2 Selfish Nodes

During the calculation of a trusted path few nodes might act selfish by going to the sleep state. These nodes will affect the trusted path calculation, as we cannot be sure the path will be stable and hence the whole concept of trusted path will get affected. So to remove these nodes the trust value of these nodes needs to be changed if it acts selfish. The trust values of these nodes needs to be affected dynamically depending on the power level of the node.

Table 1. Selfish Node Rule Base. Power Level.

Trust value(TV)	LOW	MEDIUM	HIGH
TV>0	Affect slightly	Affect highly	Affect highly
TV=0	Don't affect	Affect slightly	Affect highly
TV<0	Don't affect	Don't affect	Affect highly

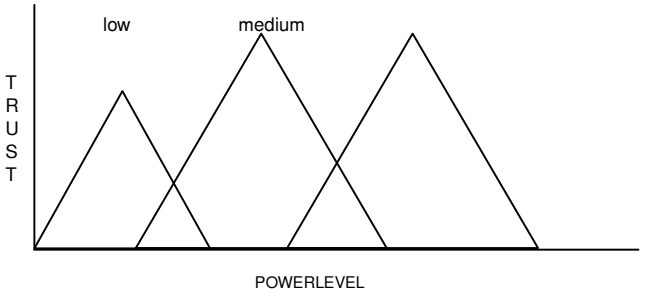

Fig. 1. Fuzzy Rule Base to determining the amount of trust to reduce based on the power level and the trust value of the node

Center of Gravity(COG) method could be used to get a crisp solution and depending on that value the DBA could decrease the trust value of the selfish node.

4.3 Group of Nodes

As seen with malicious nodes, a single node cannot affect the trust value drastically but if two or more nodes combine and work together to increase each others trust value illegally then it will result in erroneous values. So these kind of situations should be taken care of.

Table 2. Group of nodes where T_A is trustworthy. Deviation from the global trust value.

T_B	**Acceptable**	**High**	**Drastic**
Positive	$T_A(B)$	$T_A(B)$	$T_A(B)$
Negative	$T_A(B)$	$2*T_A(B)/3$	$T_A(B)/3$

Table 3. Group of nodes where T_A is trustworthy. Deviation from the global trust value.

T_B	Acceptable	High	Drastic
Positive	$T_A(B)$	$2*T_A(B)/3$	$T_A(B)/3$
Negative	$2*T_A(B)/3$	$T_A(B)/3$	0

To remove this security vulnerability we could enforce some constrains depending upon the trust values updated by the entities interacting and the frequency of trust update.

4.4 Hacking at the Network and the Database Layer

Hacking or interfering with the trust values at the transmitting or at the database should be dealt separately. It depends upon the security policies adopted in the network layer and in the database.

5 Performance Evaluation

A comparison between the Probabilistic Trust Model(PTM) and the proposed Global Trust Model(GTM) is done. The various metrics used are:

Number of nodes denotes the number of active nodes currently in contention in the environment such that all possible interactions take place only between a subset of these nodes.

Simulation time is a collection of different times such as the time taken to quantify trust according to the trust strategies and policies of the entities. It also includes the response times of the entities involving themselves in trust computation.

Throughput is defined as the number of packets sent successfully in unit time by an entity through to another entity via the communication medium. The trust value of an maliciousness of the entity therefore depends both on the maliciousness of the medium as well as the success rate of transmission of the network architecture.

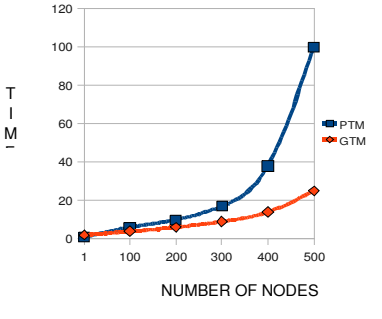

Fig. 2. Simulation time for GTM and PTM models

Number of interactions is determines by the network traffic density as well as the number of nodes in the system. The higher the number of nodes, the higher the potential number of interactions. Also, the use of recommendations may increase the number of interactions in case of higher number of nodes in PTM.

In PTM the number of transactions made to calculate the trust value depends upon the number of neighboring nodes to which the request is made. So as the number of recommendation increases to get an accurate value the network traffic also increases which results in a poor performance of the environment. In GTM the number of transactions is always a constant and is always affected by the network load.

In a PTM, we have the weighing mechanism that gives weight-age to the most recent behavior of the node. Hence, if the node is found to be malicious, then the most recent behavior outweighs the rest and communication is suspended. But the time taken to find out if the node is malicious is an overhead when a node enters the environment.

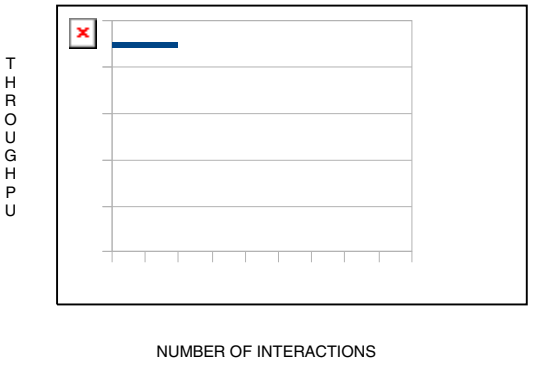

Fig. 3. Throughput of PTM versus GTM

In a GTM, this overhead is avoided because the trust value from the global data store is merged with the local data store and this value is used to decide on the threshold of trust. As a result, GTM makes a faster adjustment in cases of malicious nodes and so, there is a lower drop in the throughput- faster recovery.

6 Conclusion

In this paper, we discuss the possibility of implementing a trust management solution for a pervasive environment. In such a situation, the importance of trust as a means of security was discussed and the usage of a global and local data store was justified. We have proposed and evaluated a trust management scheme where a global data store is used along with a local data store and the combination of both the data stores in conjunction with the algorithms to compute trust. When entering a new environment where local information is not known, the weight-age is given to global data store. Also the various security issues in the GTM is also handled. This enhanced the accuracy of environment based behavior of the nodes.

References

1. Sun, T., Denko, M.K.: A Distributed Trust Management Scheme in the Pervasive Computing Environment. In: Canadian Conference on Electrical and Computer Engineering CCECE 2007, April 2007, vol. 2226, pp. 1219–1222 (2007)
2. Yuan, W., Guan, D., Lee, S., Lee, Y.: The Role of Trust in Ubiquitous Healthcare. In: 9th International Conference on eHealth Networking, Application and Services, June 2007, pp. 312–315 (2007)
3. Sun, T., Denko, M.K.: Performance Evaluation of Trust Management in Pervasive Computing. In: 22nd International Conference on Advanced Information Networking and Applications AINA 2008, March 2008, vol. 2528, pp. 386–394 (2008)
4. Wang, Y., Vassileva, J.: Bayesian NetworkBased Trust Model. In: Second International Conference on Internet Monitoring and Protection, ICIMP 2007, July 15, p. 26 (2007)
5. Black, J.P., Segmuller, W., Cohen, N., Leiba, B., Misra, A., Ebling, M.R., Stern, E.: Pervasive Computing in Health Care: Smart Spaces and Enterprise Information Systems
6. Yajun, G., Hao, C., Zhongqiang, Y., Huihui, D.: Generalized Trust Negotiation for Pervasive Computing. In: ISECS International Colloquium on Computing, Communication, Control, and Management, CCCM apos 2008, August 2008, vol. 1(34), pp. 684–687 (2008)
7. Denko, M.K., Sun, T.: Probabilistic Trust Management in Pervasive Computing. In: IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, EUC apos 2008, December 2008, vol. 2(1720), pp. 610–615 (2008)
8. Xu, W., Xin, Y., Lu, G.: A Trust Framework for Pervasive Computing Environments. In: International Conference on Wireless Communications, Networking and Mobile Computing, WiCom 2007, September 2007, (2125), pp. 2222–2225 (2007)
9. Ranganathan, K.: Trustworthy Pervasive Computing: The Hard Security Problems. In: Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, March 17, vol. 14
10. Ztoupis, D., Zarifis, K., Stavrakakis, I., Xenakis, C.: Towards a Security Framework for an Established Autonomous Network. In: 3rd International Symposium on Wireless Pervasive Computing, ISWPC 2008, May 2008, (79), pp. 749–754 (2008)
11. Kagal, L., Undercoffer, J., Joshi, A., Finin, T., Perich, F.: A Security Architecture Based on Trust Management for Pervasive Computing
12. Dong, C., Dulay, N.: Privacy Preserving Trust Negotiation for Pervasive Healthcare. In: Pervasive Health Conference and Workshops, November 29-December 1, p. 19 (2006)
13. Hassan, J., Sirisena, H., Landfeldt, B.: TrustBased fast Authentication for Multiowner Wireless Networks (Feburary 2008)
14. Khare, R., Rifkin, A.: Trust management on the world wide web. Computer Networks and ISDN Systems 30, 651–653 (1998)
15. Kim, T.K., Seo, H.S.: A Trust Model using Fuzzy Logic in Wireless Sensor Network
16. Wu, Z., Weaver, A.C.: Application of Fuzzy Logic in Federated Trust Management for Pervasive Computing