A Trust Model for Ubiquitous Systems based on Vectors of Trust Values

Hassan Jameel, Le Xuan Hung, Umar Kalim, Ali Sajjad, Sungyoung Lee and Young-Koo Lee
Department of Computer Engineering, Kyung Hee University
Sochen-ri, Giheung-eup, Yongin-si, Gyeonggi-do, 449-701, South Korea
{hassan, lxhung, umar, ali, sylee}@oslab.khu.ac.kr, yklee@khu.ac.kr

Abstract

Ubiquitous Computing foresees a massively networked world supporting a population of diverse but cooperating mobile devices where trust relationships between entities are uncertain. Though there have been lots of effort focusing on trust for Ubiquitous Systems, they did not attach enough importance to uncertainty in their model. On the other hand, most of the works draw a general picture without a detailed computational model. In this paper, we present a trust model based on the vectors of trust values of different entities. The evaluation of trust depends upon the recommendation of peer entities common to the interacting entities. These recommendations are weighted according to the number and time of past interactions. Furthermore we present a method of handling false recommendations without introducing significant computational burden. The model can calculate trust between two entities in situations both in which there is past experience among the interacting entities and in which the two entities are communicating for the first time. Several tuning parameters are suggested which can be adjusted to meet the security requirement of a ubiquitous system.

1. Introduction

Large scale distributed systems, such as the Grid, frequently require interaction between large number of different entities, e.g. processes interacting with other processes. Sometimes these entities belong to different network domains governed by different security policies. At other times two new entities may have to collaborate without any prior history of interaction. The entities would like to grant access privileges to others or allow a requested action based on some notion of trust. The traditional security mechanisms in which the resource owner confirms through an internal database after authentication to allow access to the requestor is inadequate for distributed and ubiquitous environments which are governed by uncertainty; interaction of an entity with an unknown entity is more a norm then an exception. A global trust evaluation model becomes necessary in this situation enabling the communicating parties to determine the trust for each other. A little over ten years ago, Marsh put much effort in his work about trust [13]. Since then, many trust models have been constructed for various computing paradigms such as ubiquitous computing, peer-to-peer networks, and multi-agent systems, etc [2, 15, 12, 3, 8, 14, 11, 18, 6, 16, 5]. In almost all of these works trust is accepted as a subjective notion by all researchers, which brings us to the problem: how to measure trust? Translation of this subjective concept into a language understood by computing entities is the main objective needed to be solved.

Let's see an example. Suppose Alice, a student in Lab A, wants to access another Lab B's Smart Office which is deployed using a Context-Aware Middleware for Ubiquitous Systems [7]. However, the security agent in the office doesn't allow her to access the Smart Office services since it can not recognize Alice's role. Therefore, Alice requests permission from Bob and Carol, members of Lab B, to use the services. Since they know each other pretty well, Bob's cell phone sends Alice's PDA a delegate to use the fax machine, and the copy machine while Carol only sends a delegate to use the copy machine as she is less acquainted with Alice. Based on some system policies, reputation of Bob and Carol, the security agent now grants Alice a right to access the service and use the copy machine, but not the fax machine for a certain duration of time. This example shows us the importance of trust over traditional security mechanisms in ubiquitous computing environment. Through trust computation, the system by itself can permit a stranger entity to access the services without any identity while still protects the system in secure manner.

In this paper we propose an approach for trust evaluation based on vectors of trust values to solve the difficulties mentioned above. The model requires all entities to keep the trust values for all the entities in a ubiquitous system. These values are then used to evaluate the trust between two entities who want to interact. The trust evaluation system

incorporates all the desired characteristics of a trust model. The rest of the paper is organized as follows. We briefly overview related work in Section 2. Section 3 formalizes the basic concepts in trust modeling. Then, in Section 4 we develop the building blocks of the trust model which we present in Section 5. Section 6 concludes the paper and mentions the future work.

2. Related Work

Since mid '90s the research community has outlined the key role of trust management models to develop more complex and dependable computer systems. From this, the importance of trust model was first highlighted by Blaze et al in their seminal paper [2]. Subsequently, Josang [10] presented an interesting classification of trust relationships and its implication to traditional security concepts.

Until now, several trust models have been proposed in the literature for different distributed systems [8]. For the Grid scenario, X.509 [15] and SPKI [4] seem adequate which propose a central Certificate Authority (CA) based trust model. However, there are a number of issues related to proxy/delegation certificates that are serious drawbacks of these models. A two-level trust model for Grid based on graph topology was proposed in [12]. They use different trust evaluation metrics for centralized grid domains and distributed Virtual Organizations (VO). A peer recommended trust model was proposed in [3] for ubiquitous computing systems. Their trust management scheme through recommendation lacks certain aspects such as the weighted recommendation of peers based on their prior interactions. In [8], a decentralized trust and reputation model for multi agent systems has been proposed whereas a probabilistic trust model is proposed in [17] for mobile agents. Both these models lack a fundamental requirement, i.e., very old recommendations should not be relevant in predicting the behavior of an entity. Another probabilistic trust model called the Beta Reputation System (BRS) [9] works by giving ratings about other users in the system. All these trust models can be generally categorized into probabilistic models and others in which the trust evaluation formulae are tuned to give the desired result.

In the fields of Ubiquitous Computing, research has paid much attention to build autonomous trust management as fundamental building block to design the future security framework. Up to now, research has focused mainly on the propagation and composition of trust information [14, 11, 18, 6] while paying less attention to how direct trust information is actually built. Though focused on distributed trust computation, [16, 5] face the problem of building trust from past experience. Michiardi et al [16] proposed an organic reputation-based framework to enforce collaboration in ad-hoc networks. Peer reputation is built by evaluating a mix of directly collected information, undirected feedback, and eventually multiple interaction classes.

Our trust model solves such problems by modeling and formalizing a novel and precise computation based on vectors of trust values with peer recommendation, confidence level, and history of past interaction. Moreover, we additionally include a new metric into this model that is the time-based evaluation. By doing this, peer recommendation value will have a higher weight if the number of peers common to both the interacting principals is higher. On the other hands, peer recommendations older than a threshold time interval have less weight over the others. Further more we present a method of handling false recommendations without introducing significant computational burden.

3. Basic Notions

In [1] trust is defined as: "*Generally, an entity can be said to 'trust' a second entity when it (the first entity) makes the assumption that the second entity will behave exactly as the first entity expects*".

It would be nice to formalize the notion of trust which would enable us to develop our model efficiently. The notion of an entity, which we refer to as a principal, can be defined as follows:

Definition 3.1. A user, a process or a resource which interacts or can interact with other users, processes or resources is called a **principal**.

In what follows, we denote a principal by *P* or *Q*. Every principal has its own security policy which describes different levels of access control privileges. Suppose *P* has *k* different levels of access control rights.

Definition 3.2. The **policy** $Pol_{P,k}$ of a principal *P*, having *k* levels of access control rights, is defined as the one-to-one correspondence from its security policy to the set $\{0,1,2,\ldots,k\}$.

As an example, consider a resource principal *P* giving two different types of access privileges *(k=2):read* and *write*. Thus $Pol_{P,2} = 1$ implies read access and $Pol_{P,2} = 2$ implies read and write access, whereas $Pol_{P,2} = 0$ implies no access at all. We will use the notation $Pol_{P,k}$ where ever we imply the set of privileges $\{0,1,2,\ldots,k\}$.

Definition 3.3. The **trust** of principal *P* on principal *Q* is a real number between *0* and *1*.

We denote the trust of *P* on *Q* as $t_{P,Q}$. From the definition $t_{P,Q} \in [0,1]^R$. *P completely trusts Q* if $t_{P,Q}$=1 and *completely distrusts Q* if $t_{P,Q}$=0.

Definition 3.4. For a principal *P*, a **trust mapping** denoted by m_P is a mapping from $[0,1]^R$ to its policy $Pol_{P,k}$ defined as:

$$m_P(x) = \begin{cases} k & , c_k \leq x \leq 1 \\ k-1 & , c_{k-1} \leq x < c_k \\ \vdots & \vdots \\ 1 & , c_1 \leq x < c_2 \\ 0 & , 0 \leq x < c_1 \end{cases}$$

where $x, c_1, c_2, \ldots, c_k \in [0,1]^R$

In the example above, the principal *P* might define a mapping function as:

$$m_P(x) = \begin{cases} 2 & , 0.5 \leq x \leq 1 \\ 1 & 0.2 \leq x < 0.5 \\ 0 & 0 \leq x < 0.2 \end{cases}$$

If the trust of *P* on another principal *Q* is 0.19, then $m_P(t_{P,Q}) = m_P(0.19) = 0$, implies that has no access privilege for the resource *P*. A highly secure principal could define the trust mapping such that only principals with trust value 0.9 could have the higher access rights whereas a less secure principal could set this value to 0.4. In the next section we will device a way to calculate this trust value and develop different aspects of a trust evaluation method to calculate the trust between two principals.

4. Our Trust Model

Whenever two principals want to interact, they should be able to evaluate the amount of trust on each other using some evaluation metric. This metric should include the recommendations of other principals that had past experiences with these principals; the more the experiences, the higher the weight of these recommendations. Moreover older experiences should have less impact on this evaluation. Finally the interacting principals' past experiences with each other should obviously have a say in this evaluation. These metrics are precisely developed in the following sections.

4.1. Peer Recommendation

We assume each principal in the system has its own unique identity. Suppose *n* is the total number of principals in the system. Each principle has a trust value for any other principal it interacted with before. Let $Q_1, Q_2, \ldots, Q_n$ denote the principals in the system. In this section we will model and formulate how to calculate the trust value of a principal requesting some action by asking the principal's reputation from other principals in the system. The other principals might lie and give a false recommendation for some mutual benefit. We will suppose a very reasonable assumption that principals with high trust values will not send false recommendations.

Definition 4.1. The **trust vector** of principal Q_i is defined as:

$$\overrightarrow{Q_i} = \big(t_{Q_i,Q_1}, t_{Q_i,Q_2}, \ldots, t_{Q_i,Q_{i-1}}, t_{Q_i,Q_{i+1}}, \ldots, t_{Q_i,Q_n}\big)$$

where $t_{Q_i,Q_k} = NULL$ if Q_i and Q_k have NOT interacted before, for all $1 \leq k \leq n, k \neq i$

Definition 4.2. The **peer set** of a principal Q_i denoted by S_{Q_i} is the set of all those principals Q, such that $t_{Q_i,Q} \neq NULL$

Definition 4.3. The **common peer** vectors of Q_i with Q_j are defined as:

$$\overrightarrow{C}_{Q_i,Q} = \big(t_{Q_i,Q_{k_1}}, t_{Q_i,Q_{k_2}}, \ldots, t_{Q_i,Q_{k_m}}\big)$$
$$\overrightarrow{C}_{Q,Q_i} = \big(t_{Q_{k_1},Q_i}, t_{Q_{k_2},Q_i}, \ldots, t_{Q_{k_m},Q_i}\big)$$
$$where \quad \{Q_{k_1}, Q_{k_2}, \ldots, Q_{k_m}\} = S_{Q_i} \cap S_{Q_j}$$

The common peer vectors for Q_j are defined likewise. Notice that the first vector represents the trust values of Q_i for all the peer principals common to both Q_i and Q_j, and the second vector represents the trust of the common peer principals on Q_i. Next, we define the peer recommendation for two interacting principals.

Definition 4.4. The **peer recommendation** for the interaction with Q_j to Q_i is defined as:

$$PR_{Q_i,Q_j} = \begin{cases} \frac{\overrightarrow{C}_{Q_i,Q} \bullet \overrightarrow{C}_{Q,Q_j}}{m} & , S_{Q_i} \cap S_{Q_j} \neq \phi \\ 0 & , S_{Q_i} \cap S_{Q_j} = \phi \end{cases}$$

Where,$m = \big|S_{Q_i} \cap S_{Q_j}\big|$ and $\overrightarrow{C}_{Q_i,Q} \bullet \overrightarrow{C}_{Q,Q_j}$ is the dot product of the common peer vectors. The peer recommendation to Q_j for the interaction with Q_i is defined similarly. We have peer recommendation will weight the recommendations on the basis of the principals trust of the common peers and the common peers' trust of the other principal.

Preposition 4.1. For any two principals $Q_i$and Q_j, $0 \leq PR_{Q_i,Q_j} \leq 1$.

Proof. The elements of the vectors $\overrightarrow{C}_{Q_i,Q}$ and $\overrightarrow{C}_{Q,Q_j}$ are trust values which according to Definition 3.3 have a value between 0 and 1. Thus $\overrightarrow{C}_{Q_i,Q} \bullet \overrightarrow{C}_{Q,Q_j} \leq m \Rightarrow 0 \leq PR_{Q_i,Q_j} \leq 1$.

Example 4.1. Suppose principals $Q_1$and Q_2 have two peers in the set $S_{Q_1} \cap S_{Q_2}$ namely $Q_3$and Q_4 with $t_{Q_1,Q_3} = 0.3$, $t_{Q_1,Q_4} = 0.8$, $t_{Q_2,Q_3} = 0.9$ and $t_{Q_2,Q_4} = 0.2$.

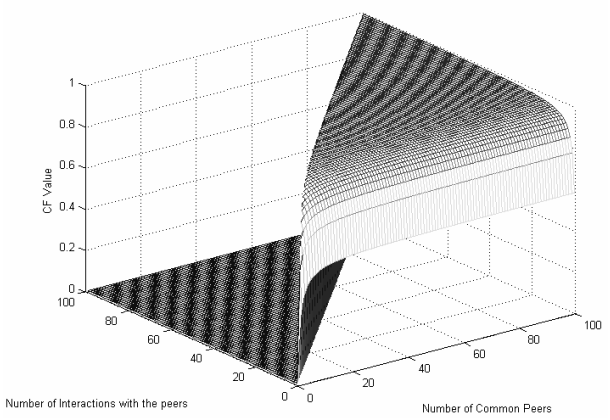

Figure 1. Confidence value CF with different values of the parameters m and I_Q with α =0.2

Then $PR_{Q_1,Q_2} = ((0.3)(0.9) + (0.8)(0.2))/2 = 0.215$

This peer recommendation can be used to calculate how much trust the two principals can put upon each other. The peer recommendation will be higher if the peers have more trust on the principals and vice versa. Thus gives a good idea about the reputation of the two principals. Notice that this value is the same for both the interacting principals. The peer recommendation involves a dot product of vector elements, one of which is the trust that the principal Q_i has on the other one and the second one is the trust that the other principal has on Q_j Thus if Q_i has a low trust value for that principal, then its recommendation will be highly minimized. Consequently, based on our assumption, a principal who gives a false recommendation about Q_j will not get any advantage as it will have a low trust value.

4.2. Confidence

Intuitively, the *PR* value calculated above should have a higher weight if the number of peers common to both the interacting principals is higher. Likewise principals with more interactions with a particular principal should have a higher say in recommendation. This introduces the notion of confidence over the *PR* value. The confidence level should be a maximum if the number of common peers and the number of individual interactions of these peers are greater than a threshold value. The function:

$$f(x) = 1 - \frac{1}{x + \alpha}$$

has the desirable property that with increasing x (x could be any positive integer) the function quickly approaches 1 and can be used to calculate this metric. α is an adjustable positive constant in the system and can be tuned accordingly. Notice that instead of the above function we could have used any other function that has the property of quickly approaching '1' with increase in the argument. Our choice of the above function is there for brevity and ease of calculation. Let I_{Q_i} and I_{Q_j} denote the total number of interactions of principals Q_i and Q_j with all the principals in $S_{Q_i} \cap S_{Q_j}$. We define the confidence (*CF*) on the *PR* value for the principals Q_i and Q_j as:

$$CF_{Q_i,Q_j} = \frac{1}{2}\left(f(m) + f(I_{Q_i})\right)$$
$$CF_{Q_j,Q_i} = \frac{1}{2}\left(f(m) + f\left(I_{Q_j}\right)\right)$$

where as before, $m = \left|S_{Q_i} \cap S_{Q_j}\right|$ The function $f(x)$ approaches 1 as x becomes bigger. Thus the *CF* value will be close to 1, if 1) the number of peers common to both the principals becomes higher and 2) the number of past interactions with these peers becomes higher. Thus the confidence is a mean of these two factors.

Example 4.2. Suppose $m = \left|S_{Q_i} \cap S_{Q_j}\right| = 5$, $I_{Q_i} = 15$, $I_{Q_j} = 5$ and $\alpha = 0.2$. Then the CF_{Q_i} value is 0.87 and CF_{Q_i} is 0.8 which shows that Q_i can be about 90% confident about the *PR* value where as Q_j is only 80% confident.

Figure 1 shows the values for *CF* with different values of the parameters m and I_Q with α =0.2. The trend being that with an increasing number of common peers and number of interactions with these peers, the confidence value approaches 1 rapidly. Figure 2 shows the change in the confidence value with different values of the adjustable constant α. Its value can be made higher if only a few number of peers are deemed necessary to increase the confidence in the *PR* value and vice versa.

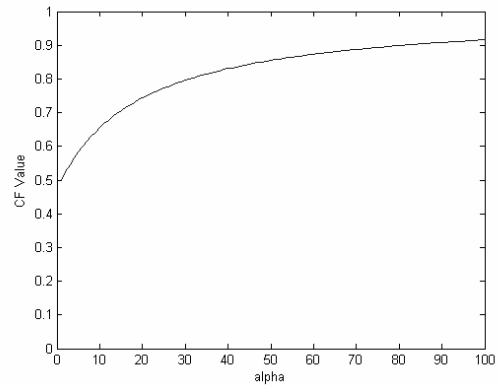

Figure 2. CF against α with $m = I_Q$ =2

4.3. History of Past Interactions

In section 3, we gave the definition of trust as the confidence of one principal on another that it will behave exactly

the same as it expects. An important factor in deciding this confidence is the history of the past interactions. Two interacting principals should keep in mind their past experiences when calculating the trust value. We can generically define successful and unsuccessful interactions between two principals based on their past behaviors, where an unsuccessful interaction means that the principal has betrayed the trust bestowed upon it. The nature of an interaction might reflect more than just a successful and unsuccessful interaction. For example, a principal might behave totally contrary to the expectations whereas another one might diverge to a lesser extent. However, as this transition is really cumbersome to model and might differ from every principal's perspective, we restrict ourselves to the two outcomes; successful and unsuccessful. Furthermore, the outcome of an interaction might be different in the view of the two principals. What one conceives as a success, the other might regard as a failure. Let us define SI_{Q_i,Q_j} as the number of successful past interactions with in the eyes of Q_i and UI_{Q_i,Q_j} as the number of unsuccessful interactions. The view of Q_j is defined likewise. We would like to give more weight to *UI* over *SI* as even a single unsuccessful interaction will certainly shatter the confidence of one principal over the other.

Definition 4.5. The **history of interaction** of Q_j as calculated by Q_i is defined as:

$$h_{Q_i,Q_j} = \max\left\{w_S SI_{Q_i,Q_j} - w_U UI_{Q_i,Q_j}, 0\right\}$$

where w_S and w_U are positive numbers; the corresponding weights of SI_{Q_i,Q_j} and UI_{Q_i,Q_j}.

Definition 4.6. The Past Interaction Evaluation (*PI*) of Q_j as calculated by Q_i is defined as:

$$PI_{Q_i,Q_j} = 1 - \frac{1}{h_{Q_i,Q_j}+1} = f\left(h_{Q_i,Q_j}\right), \alpha = 1$$

4.4. Time based evaluation

Intuitively, very old experiences of peers should have less weight in peer recommendation over new ones. In other words, peer recommendations older than a threshold time interval should have less weight over the others. We can put this desired property in our evaluation model if every principal keeps a time stamp with its latest interaction with every other principal. We denote the time stamp between principals *P* and *Q* as $\tau_{P,Q}$. Further, let $\Delta\tau$ denote the threshold time interval. Now, suppose Q_i and Q_j decide to interact at time τ. We define the time based evaluation (*TE*) for both Q_i and Q_j as:

$$TE_{Q_i,Q_j} = \frac{m}{\sum_{l=1}^{m}\left\lceil \Delta\tau_{Q_j,Q_{k_l}} / \Delta\tau \right\rceil}$$
$$TE_{Q_i,Q_j} = \frac{m}{\sum_{l=1}^{m}\left\lceil \Delta\tau_{Q_j,Q_{k_l}} / \Delta\tau \right\rceil}$$

where $\Delta\tau_{X,Y} = \tau - \tau_{X,Y}$, $\{Q_{k_1}, Q_{k_2}, \ldots, Q_{k_m}\} = S_{Q_i} \cap S_{Q_j}$ and $m = \left|S_{Q_i} \cap S_{Q_j}\right|$

It is clear that if all the recommendations are within the interval $\Delta\tau$ then *TE* will have a value equal to 1.

5. Trust Evaluation Metric

Based on the aforementioned metrics, we are now ready to describe our trust evaluation metric. The trust metric is defined as a weighted arithmetic mean of *PR*, *CF*, *TE*, and *PI*. More precisely, the trust between two principals Q_i and Q_j who want to interact can be calculated as:

$$t_{Q_i,Q_j} = \frac{w_1\left(PR_{Q_i,Q_j}\right)\left(\frac{CF_{Q_i,Q_j}+TE_{Q_i,Q_j}}{2}\right)+w_2\left(PI_{Q_i,Q_j}\right)}{\sum_{i=1}^{2} w_i}$$

where $f_i \in N$ and they can be adjusted to a suitable value if more weight is to be given to a specific metric. For example, past interactions evaluation should be given more importance over the others. The *PR* value is weighted over *CF* and *TE*.

Example 5.1. Suppose Q_i and Q_j wish to interact and they calculate *PR*=0.215 , *CF*=0.9, *TE*=1, and *PI*=0 as they never interacted before. Assuming $f_1 = 1$ and $f_2 = 4$ we get $t_{Q_i,Q_j} = 0.04$. However, if they have successfully interacted once, then the trust value is 0.44 which reaches $t_{Q_i,Q_j} = 0.84$ with 20 successful interactions without any unsuccessful interaction keeping the other metrics the same. Figure 3 shows the growth of trust value with increasing *PR* and *PI*=0.

The history of past interactions between the two interacting principals has a great impact on trust calculation. Naturally, if the interacting principals had bad experiences with each other, they will be less willing to allow a requested access or action. With increasing number of successful interactions, a principal's trust value in the whole model increases. If a new principal joins the system without any prior interaction with all other principals in the system, the other principals can have a choice whether to give any privilege to this principal or not. They could give a minimum trust value to this entity or a highly secure principal might not give any access at all for a total stranger. Notice that to encounter false recommendations we have assumed that

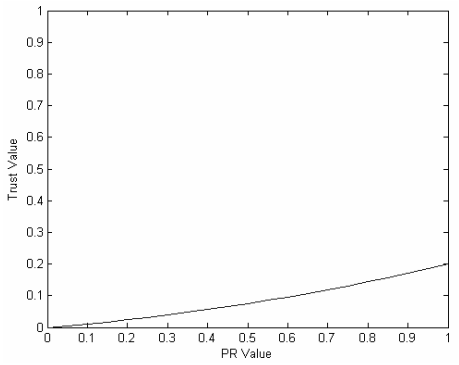

Figure 3. The trust value will not increase above a certain level if the number of unsuccessful interactions is large enough to make PI=0

only principals with low trust values will give false recommendations about a certain entity. In this case, the calculated peer recommendation value will have less weight in the dot product.

6. Conclusion and Future Work

In this paper, we present a model for trust based on the vectors of trust values of different entities in ubiquitous computing. Distinguished from previous trust model, our trust model takes uncertainty of trust into account with a precise computation model. Besides basic factors of trust computation such as peer reputation, confidence, and history of past interaction. We additionally include time based evaluation factor to calculate trust value and efficiently handle false recommendations. The calculation of the trust depends upon the recommendation of peer entities common to the entities which are weighted according to the number of past interactions and the time of last interaction. The model can calculate trust between two entities in situations both in which there is past experience among the interacting entities and in which the two entities are communicating for the first time. Several tuning parameters are suggested which can be adjusted to meet the security requirement of a distributed system. A highly secure system can adjust these parameters such that only a few entities with very high reputation and recommendation are allowed to perform requested actions.

As a future work we will implement the proposed trust model and use it in a ubiquitous environment. Also, we will add on risk analysis into our model, we believe security measures must be proportional and appropriate for the risk involved: a user may happily distribute a business card to strangers to advertise their business, but may be quite careful as to whom they give their mobile phone number. Another future work is how to exchange vectors among principals so that certain principals can enforce their trust value on the others.

References

[1] Itu-t recommendation x.509 (2000 e). information technology. open systems interconnection-the directory: Public-key and attribute certificate frameworks.

[2] J. Blaze, M. Feigenbaum. Decentralized trust management. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pages 164–173, 1996.

[3] M. Carbone. A formal model for trust in dynamic networks. In *International Conference on Software Engineering and Formal Methods (SEFM'03)*. IEEE, 2003.

[4] C. Ellison. Spki certificate theory. *Internet Request for Comments: 2693*, 1999.

[5] S. Ganeriwal and M. B. Srivastava. Reputation-based framework for high integrity sensor networks. In *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 66–77. ACM Press, 2004.

[6] R. Guha, R. Kumar. Propagation of trust and distrust. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 403–412. ACM Press, 2004.

[7] S. Hung, N. Q. Kiani. Developing context-aware ubiquitous computing systems with a unified middleware framework. In *2004 International Conference on Embedded and Ubiquitous Computing*. Springer Verlag, 2004.

[8] N. Huynh, T.D. Jennings. Developing an integrated trust and reputation model for open multi-agent systems. In *AAMAS-04 Workshop on Trust in Agent Societies*, 2004.

[9] R. Ismail. The beta reputation system. In *Proceedings of the 15th Bled Conference on Electronic Commerce*, 2002.

[10] A. Josang. The right type of trust for distributed systems. In *New security paradigms workshop*, pages 119–131, 1996.

[11] M. Kamvar, S.D. Schlosser. The eigentrust algorithm for reputation management in p2p networks. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 640–651. ACM Press, 2003.

[12] H. Li, T.Y. Zhu. A novel two-level trust model for grid. In *ICICS*, pages 214–225, 2003.

[13] S. Marsh. Formalising trust as computational concepts. *Ph.D Thesis, University of Stirling*, 1994.

[14] R. Matthew, R. Agrawal. Trust management for the semantic web, 2003.

[15] S. Mendes. A new approach to the x.509 framework: Allowing a global authentication infrastructure without a global trust model. In *Proceedings of NDSS 95*, 1995.

[16] R. Michiardi, P. Molva. Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security*, pages 107–121. Kluwer, B.V., 2002.

[17] N. Patel, J. Jennings. A probabilistic trust model for handling inaccurate reputation sources. In *iTrust*, pages 193–209, 2005.

[18] J. Theodorakopoulos, G. Baras. Trust evaluation in ad-hoc networks. In *WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security*, pages 1–10. ACM Press, 2004.