A Flexible Trust-Based Access Control Mechanism for Security and Privacy Enhancement in Ubiquitous Systems

Pho Duc Giang[1], Le Xuan Hung[1], Sungyoung Lee[1], Young-Koo Lee[1] and Heejo Lee[2]
[1]*Computer Engineering Department, Kyung Hee University, Korea*
[2]*Computer Science and Engineering Department, Korea University, Korea*
{pdgiang, lxhung, sylee}@oslab.khu.ac.kr, yklee@khu.ac.kr, heejo@korea.ac.kr

Abstract

It is the ubiquity and mobility absolutely necessary for ubiquitous computing environments that raise new challenges for pervasive service provision invisibly. Particularly, mobility of users/devices causes un-predefined and unpredictable changes in physical location and in available resources and services, event at runtime and during the same service session, thus forcing us to consider very dynamic aspects of evaluation when designing an access control mechanism. Alternatively, there is generally no a priori trust relationship among entities interacting in pervasive computing environments which makes it essential to establish trust from scratch. This task becomes extremely challenging when it is simultaneously necessary to protect the privacy of the users involved. In this paper[*]*, we first show how trust evaluation process of the user's system can be based on previous accesses and peer recommendations. A solution then relied on trust to control access is proposed that depends upon pre-defined access control security policy. Several tuning parameters and options are suggested so that end-users can customize to meet the security and privacy requirement of a ubiquitous system.*

1. Introduction

The vision of ubiquitous computing, with devices seamlessly integrated into the life of everyday users, and services readily available to users anywhere anytime [1,2] is becoming now a reality. However, the flexibility of the environment comes at a cost – higher risks and privacy disclosures. In this environment, much of the user context which is constantly being captured, transmitted, stored by wireless devices (sensors, access points, etc) usually contains privacy-sensitive information, protecting the private resources is of serious interest to end-users.

Additionally, the environment itself lacks a priori trust among parties and the interactions are ad hoc naturally. In other words, trust relationships have to be started from scratch. The traditional association with a network provider may not exist, replaced by a far more vague connection with a number of unknown entities, network nodes and service providers. Therefore, designing a sufficient and suitable access control mechanism for security and privacy in ubiquitous computing environments becomes an urgent demand.

Furthermore, in ubiquitous world, users' access rights change dynamically with respect to their relationship with the medium by which data are generated and sometimes the clients cannot be predetermined. Traditional authentication and access control are effective only in situations where the system knows in advance which users are going to access and what their access rights are. Regarding that point, we need a flexible solution capable of control the security and privacy issues on the runtime so as to provide essential amount of services to requesters who are either unfamiliar with the system or do not have enough access rights to certain services.

In this paper, we introduce the idea of using trust to provide finer-grained access control over the sensitive resources, thus helping to manage the security and privacy issues efficiently. The usability goal is where the end-user should do nothing to log in, she should simply use services/resources, and these services believe the user based on her trust level. In order to determine whether someone is trusted or not to allow her access different parts of our services, we first depend upon two different evaluation factors: peer

[*] This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement) (IITA-2006-C1090-0602-0002).
Dr. Sungyoung Lee is the corresponding author.

2007 International Conference on Multimedia and Ubiquitous Engineering(MUE'07)
0-7695-2777-9/07 $20.00 © 2007 IEEE

recommendation, and time-based past access history to calculate the trust value. After that, based on the outcome of trust estimation process, we assign one of the two possible access permissions: *allow* or *block* to the requester. By applying pre-defined trust-based security access control policies, we are able to administer and disseminate appropriate services/resources to the partner.

The remaining paper is organized as follows. We briefly overview related work in Section 2. Next, in Section 3, we formalize fundamental concepts of trust to elaborate the functional aspects of the scheme proposed in Section 4. We are then describing the methodology in detail in Section 4. Finally, in Section 5, conclusions and future work are drawn.

2. Related Work

Many of related research activities [3,4,5] have been focused on how to efficiently protect data from unauthorized access and even more study is still in progress. Also, quite much research work has been dedicated to access control mechanism and its connected application [6,7,8,9]. However, these efforts have focused on identity based access control or use implicit trust to grant access and just few works about access control deploying explicit trust.

Conventional access control methods such as mandatory access control (MAC) and discretionary access control (DAC), delegate or revoke users' access privilege directly. However, due to the problem that MAC and DAC mechanisms assign a security clearance to each user to restrain access capability, these systems will become inconvenient and complicated when the number of users and the relationship among them increase rapidly.

Role Based Access Control (RBAC) [8,9] is probably one of the best known methods for access control, where entities are assigned roles in which permissions associated with each role, instead of users. Unfortunately, this is difficult for systems where it is not possible to assign roles to all users and in the situation that foreign users are common.

Pirzada et al. [10] extends Kerberos protocol for mobile ad hoc network security authentication by deploying multiple Kerberos servers for distributed authentication and load distribution. All servers share a secret key, and copy the other users' hashed password periodically or on demand. Their solution overcomes the single point failure created by central key server of the traditional Kerberos model. Nevertheless, authentication is relied upon users' identity. Thus, it more or less affects the privacy aspects of the entities joining the environment.

Kagal et al. [11] applied trust factor which is based on time-lived signed delegations and XML signatures (www.w3.org/signature) to examine unfamiliar requests before making a decision whether those requests should be allowed to access certain service or not. They also mentioned about the issue that a stranger who wishes to access some resource should find privileged users for asking delegation. However, they have not shown how and which evaluation method the stranger should be trusted properly.

Recently, Gua Ya-Jun et al. [12] have presented a trust-based access control model to secure ubiquitous computing application. They proposed a resource-constrained trust negotiation to establish initial trust for authenticating strangers. Still, their solution basically extends the RBAC so it faces with the inherent drawbacks of the RBAC model.

In the field of Ubiquitous Computing, the large scale deployment of pervasive computing applications heavily depends on the assurance of essential security and privacy properties for users and service providers. In addition to security exposures due to the underlying mobile and wireless communications, pervasive computing applications bring up new privacy issues. In this study, we show how it is possible to establish trust for security and privacy enhancement by tackling two main security problems of pervasive computing. First of all, such environments lack a priori trust among parties. In other words, trust relationships have to be started from scratch. We propose a trust evaluation model which involves precise computation to update the decisions dynamically. Furthermore, it is important to ensure that intrusive technology cannot spy users by tracing them and by recording their acts. Thus, we also design a trust-based access control policy depending upon particular services and resources which could be better appropriate to the ubiquitous environment.

3. Fundamental Concepts

It would be useful to formalize the notion of trust which enables us to develop the solution efficiently. Trust is an area of study in which people with different backgrounds have tried to base their own views on their own circumstances. Out of several definitions of trust, one definition that we would like to mention here is by Grambetta [13]. Grambetta relates trust to future expectation. He defines trust as a probability of a trusted entity doing something beneficial for the trusting entity. In other words, if Bob does something which Alice expects (assuming that this expectation is driven by the fact that the result will be beneficial to Alice) then Alice can trust Bob. If the result is not what Alice expects then Alice cannot trust Bob. Regarding

the notion of a requester, which we refer to as a principal, it can be formally defined as follows:

Definition 3.1 A user, a service, an application, or a system which requests or can send requests to other users, services, applications, or systems is called a principal.

We denote a principal by P or Q for the rest of this paper. In our proposed approach, every principal has its own trust-based access control policy which indicates different types of resources to be disclosed.

Definition 3.2 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 $T_{P,Q}$. According to the definition, $T_{P,Q} \in [0,1]$. Hence, P completely trusts Q if $T_{P,Q} = 1$ and completely distrusts Q if $T_{P,Q} = 0$.

Definition 3.3 The access control policy $P_{P,k}$ of a principal P, having k types of resources to be shared, is defined as a mapping from its policy to the set of actions {A - Allow, B - Block}.

Assume that a principal P provides two different types of resources (k = 2). Hence, $P_{P,2}$ = A implies full access to 2nd resource and $P_{P,1}$ = B implies no access to 1st resource at all.

Definition 3.4 For a principal P, a **trust-access mapping** denoted by M_P is a mapping from [0,1] to its access control policy $P_{P,k}$ defined as:

$$M_P(x) = \begin{cases} A & , c_k \le x \le 1 \\ A & , c_{k-1} \le x \le 1 \\ \vdots & \vdots \\ A & , c_2 \le x \le 1 \\ A & , c_1 \le x \le 1 \end{cases} \Leftrightarrow$$

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

Where $x, c_1, c_2, \ldots, c_k \in [0,1]$.

In the previous example, the principle P might define a mapping function as:

$$M_P(x) = \begin{cases} A & , 0.55 \le x \le 1 \\ & \\ A & , 0.70 \le x \le 1 \end{cases} \Leftrightarrow$$

$$M_P(x) = \begin{cases} B & , 0 \le x < 0.55 \quad (2) \\ & \\ B & , 0 \le x < 0.70 \quad (1) \end{cases}$$

If the trust evaluation of P on another principal Q which wishes to access P's 1st resource is just 0.2, then respecting (1) and even (2) $M_P(T_{P,Q}) = M_P(0.2)$ = B (Block), implies that P has no resource exposure for the request Q. In other words, if Q requests for access 1st resource of P, Q will not receive any related information since the Q's trust value is unacceptably low. In the next section we will technically present a procedure to evaluate the trust value and develop different aspects of a trust evaluation method to calculate the trust of any principal.

4. Our Approach

In this section, we propose an access control scheme based on the concept of trust with peer recommendation and past access history, and the trust-based security policy to guarantee that users' resources will not be delivered in a wrong way to a wrongdoer. There are two different stages in our solution: i) we estimate the trust value for each request coming from an entity; ii) we exploit the trust-based policy to make decision whether to accept the request or not. All these two phases can be performed automatically. We aimed to develop a system that required minimal ongoing user involvement. In particular, we did not want users to have to repeatedly evaluate the acceptability of a request for private resources. Instead, we wanted to push a query's acceptance or rejection to the system itself and only bring a query to users' consideration if they had not established a policy to handle it. Moreover, we believe users' resources/services should be protected by default; as a consequence, the system architecture lets a user elect to share certain resource rather than protect specific one.

4.1. Trust Evaluation Module

Suppose that ubiquitous systems, like smart offices, deploys our context-aware middleware CAMUS [14] servers to provide appropriate services for clients. In this situation, some rooms are multi-purpose, and are designed for internal meetings as well as for business conferences in which stricter access control should be required. Thus, the process of the system P to evaluate the trust value of any entity Q is as follows: CAMUS servers receive a request Q which inquires to get the current available service. At the beginning, P examines the query to determine whether the request comes from a common source or not. If it is from a familiar starting point, the servers will transfer the query to the trust calculation module. This module will initially base on the past access history stored in log-files of the servers during specific time interval to produce proper trust value for the request. If there is no any previous interaction correspondent to this query, now this module will ask other trusted entities who are currently active in a certain range of this smart environment to give recommendations for Q. The general flow of trust evaluation is shown in Figure 1.

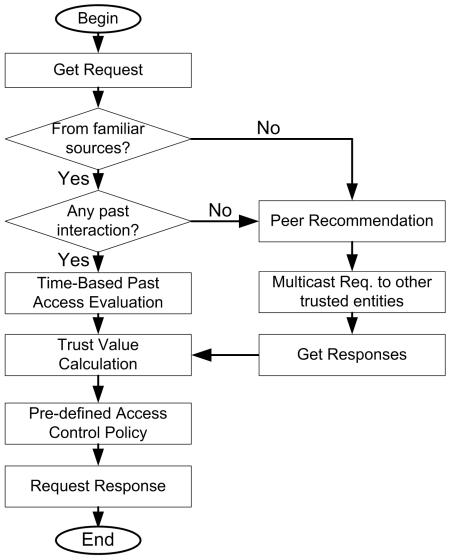

Figure 1. Flow chart of trust evaluation

4.1.1. Time-Based Past Access History. Past Access History is an entity's previous interaction knowledge to certain principal. As a matter of fact, past access history is usually recorded in log files on the subjects' systems that keep track of all actions relational participants took with the system. Since the log file is configured to keep monitoring events for a specified amount of time, it is reasonable for us to apply trust evaluation based on the temporal factor.

We can generally define successful and unsuccessful access between a principal Q and a system P established on the past behaviors in which an unsuccessful access means that the principal did not get the outcome as it expected. Let us define SA_t as the number of successful past access times and UA_t as the number of unsuccessful access times of the system at time t. Now, the trust value of Q as calculated by a system P is defined as follows:

$$T_{P,Q} = \left[\frac{SA_t}{SA_t + UA_t}\right]\left[1 - \frac{1}{Ae^{(\alpha SA_t - \beta UA_t)}}\right]$$

Where α, β, and A are adjustable positive constants in the system and can be tuned if necessary.

The expression $\left[1 - \frac{1}{Ae^{(\alpha SA_t - \beta UA_t)}}\right]$ approaches '1' quickly with an increase in the number of Successful Access times and/or a decrease in the number of Unsuccessful Access times within certain period of time. Notice that our choice of the above expression is for the smooth property of the exponential function and ease of calculation. It turns out that $T_{P,Q} = 0$ if $(\alpha SA_t - \beta UA_t) < 0$. In other words, the trust value of principal Q is equal to 0 if its number of Unsuccessful Accesses is greater than the number of Successful Access times of the system P. The factor $\left[\frac{SA_t}{SA_t + UA_t}\right]$ indicates the percentage of successful interactions in the whole communication session. We actually exploit the time-based sliding window mechanism [15] to estimate the percentage of successful communications.

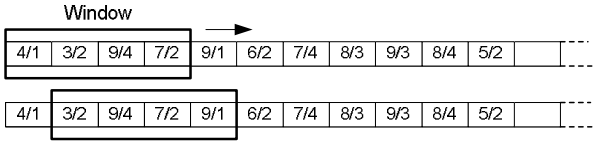

Figure 2. Time-Based Sliding window mechanism

A sliding window is a variable-duration window that allows the system to compute different principals' trust value relied on the number of successful access times in a specified number of timing units. Note that the window size could be changed depending on the user's configuration. In Figure 2, the current window length is presumably configured as a 4-unit sliding window. During the first timing interaction unit, the number of successful and unsuccessful accesses was 4 and 1 respectively. Once a unit of time passes, the window slides one time unit from left to right, eliminating the previous interactions in the first unit from the trust calculation. Hence, very old past history information will not be involved in working out a trust evaluation as time goes by. Under the simple example shown in Figure 2 with α = 1, β = 2, and A = 1,

$$T_{P,Q} = \left[\frac{23}{(23+9)}\right]\left[1 - \frac{1}{e^{(1.23-2.9)}}\right] = \frac{23}{32}\left[1 - \frac{1}{e^5}\right] \approx 0.70$$

for the first interval. However, $T_{P,Q}$ will be changed in the next interaction interval since the number of successful and unsuccessful access times are 9 and 1 which are slightly different from the previous ones:

$$T_{P,Q} = \left[\frac{28}{(28+9)}\right]\left[1 - \frac{1}{e^{(1.28-2.9)}}\right] = \frac{28}{37}\left[1 - \frac{1}{e^{10}}\right] \approx 0.76.$$

4.1.2. Peer Recommendation. Peer Recommendation factor is required when the system has no or not enough information about a principal. Obviously, if there exists certain peer having more interactions with this principal, his suggestion should be likely logical and important for assessing the trust value.

Assume that the system was not familiar with this kind of request before so our system P has to ask other peers in the environment for their suggestions. In this situation, the system will send multicast a request for comments about the new principal Q to its confident community. We denote the time stamp between a principal Q and the system P as $\tau_{P,Q}$ and τ is the time at which Q decides to interact with P. Suppose n is the number of principals currently active in the

environment. Let $P_1, P_2, \ldots, P_n$ represent the principals in the space. We also say that principals with high trust values will not send false recommendations. Moreover, let $\Delta\tau$ denote the threshold time interval. Under those assumptions, definition 3.2, and Figure 3, the trust value for the requesting principal Q is defined as follows:

$$T_{P,Q}=\frac{\eta_1 T_{P,P_1} T_{P_1,Q}+\eta_2 T_{P,P_2} T_{P_2,Q}+\eta_3 T_{P,P_3} T_{P_3,Q}+\ldots+\eta_n T_{P,P_n} T_{P_n,Q}}{n} \quad (n \neq 0)$$

$$\Leftrightarrow T_{P,Q}=\frac{\sum_{i=1}^{n} \eta_i T_{P,P_i} T_{P_i,Q}}{n} \quad (n \neq 0)$$

Where $\eta_i = Be^{\theta^{\frac{\Delta\tau_{P_i,Q}}{\Delta\tau}}} \in (0,1]$, with $\Delta\tau_{P_i,Q} = \tau_{P_i,Q} - \tau$. B and θ are adaptable positive constants which can be chosen apart to guarantee that $\eta_i \leq 1$. For example, we select θ = 1. To establish $\eta_i \leq 1$, B must be picked out such that $B \in (0, \frac{1}{e^{\frac{\Delta\tau_{P_i,Q}}{\Delta\tau}}}]$. Since $\Delta\tau_{P_i,Q} \leq \Delta\tau$, we have $B_{max} \approx 0.46$. Obviously, $T_{P,Q} = 0$ if $n = 0$. In other words, peer recommendation will not be involved in trust evaluation process if there is no peer in the space. Besides, notice that η_i swiftly approaches '1' with increase in the argument $\Delta\tau_{P_i,Q}$. This means that very old and short experiences of peers with the principal in a period of time $\Delta\tau$ should have less weight in trust estimation than the new and long ones. In Figure 4, we show that the value of η_i increases quickly if $\Delta\tau_{P_i,Q}$ augments gradually within 100 timing units. After finishing the trust evaluation phase, we move towards the second phase in order to decide whether to deliver protected resources to the principal (Figure 1).

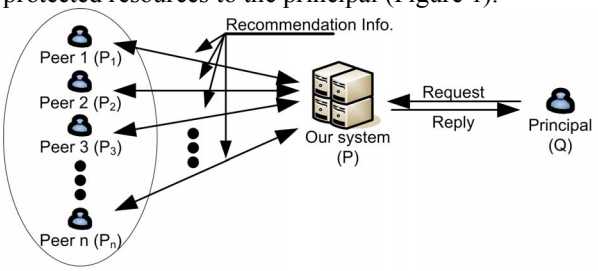

Figure 3. A Peer Recommendation Scenario

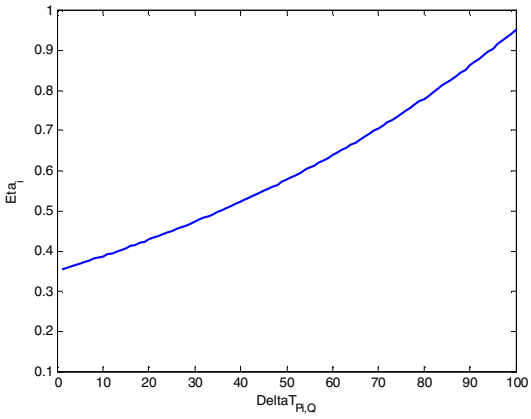

Figure 4. η_i against $\Delta T_{Pi,Q}$ with θ = 1 and B = 0.35

4.2. Trust-Based Access Control Policy

We design a Trust-based Access Control Policy (TACP) module to describe the constraints such that the end-user's resources are shared in the manner that she would expect. Requesters cannot directly access available resources/services, but get a reference to the TACP. Whenever a principal asks to access a resource, the trust evaluation module intercepts its request and estimates its trust value. Once a principal's trust level was quantized by our system, it will be considered as one of two pre-defined states: *Allow* or *Block* with the support of a trust-privacy mapping (definition 3.4) according to specific resource.

$$M_P(x) = \begin{cases} \text{Allow} & , c_1 \leq x \leq 1 \\ \text{Block} & , 0 \leq x < c_1 \end{cases}$$

Where c_1 is an adjustable positive constant and can be tuned accordingly. We consider the case of Alice's ubiquitous supported smart office in which different resources/services, such as printers, fax machines, storage servers, etc are available for sharing. When Alice hosts a teleconference to present the company proposals to her colleagues, she takes her own Pocket PC to access her office's resources, retrieving the necessary files and programs. Once the conference is established, the ad hoc group can also share applications and use a common resource like some ftp server located at Alice's place to upload/download material but not others. This scenario raises access control policy need. On the one hand, Alice's resources have to be protected from illicit accesses from unauthorized members; on the other hand, local resources/services have to be secured from attendees' unauthorized actions. So as to accomplish access control successfully, we show an example of particular access control policy as in Table1.

Table 1. The content of an entry in an access control policy

Order	Resources/Services	Trust Value Threshold	Action	Comment
01	Printer01	0.35	Allow	Alice's Printer01
02	Fax_Machine	0.45	Allow	Alice's Fax Machine
03	FTP_Server01	0.75	Allow	Alice's FTP server
04	Storage_Server01	0.80	Allow	Alice's document
05	Storage_Server02	0.90	Allow	Alice's personal data
06	Any	Any	Deny	Any

5. Conclusion and Future Work

The aim of this paper is to contribute to the development of a strict discipline for designing access control mechanisms in ubiquitous environments. In this study, we introduce a trust-based access control model by taking uncertainty of trust into account with a precise computation model. Additionally, we apply customizable access control policy to efficiently handle malicious principals. The calculation of trust depends on the time of last accesses and peer reputation common to the entities. Besides, several tuning parameters and options are suggested which can be technically adapted to meet the requirements of a pervasive computing space. A highly secure and private system can fit these variables such that only a small number of principals with appropriate reputation and recommendation are allowed to gain sensitive resources.

Eventually, we believe that there is lots of work to do in the implementation area. As a future work we are going to build up the proposed trust evaluation and access control policy modules that put our findings into practice, allowing people to differentiate exposure their resources by trust estimation.

References

[1] M. Weiser, "The Computer for the 21st Century," Scientific America, pp. 94-104, Sept. 1991; reprinted in IEEE Pervasive Computing, pp. 19-25, Jan.-Mar. 2002.

[2] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges," IEEE Personal Communications, pp. 10-17, Aug. 2001.

[3] G. Appenzeller, M. Roussopoulos, and M. Baker, "User-friendly access control for public network ports," in Proc. IEEE INFOCOM, pp. 699–707, 1999.

[4] R. Sailer and J.R. Giles, "Pervasive authentication domains for automatic pervasive device authorization," in Proc. Second IEEE Annual Conference, pp. 144-148, 14-17 March 2004.

[5] T. Yu and M. Winslett, "A Unified Scheme for Resource Protecion in Automated Trust Negotiation," in Proc. IEEE Symp. Security and Privacy, pp.110-122, May 2003.

[6] M.J. Moyer, and M. Ahamad, "Generalized Role-Based Access Control," in Proc. 21st International Conference on Distributed Computing Systems, IEEE Computer Society Press, pp. 391-398, 2001.

[7] M. J. Covington, W. Long, S. Srinivasan, A. K. Dey, M. Ahamad and G.D. Abowd, "Securing Context-Aware Applications Using Environment Roles," in Proc. 6th ACM Symposium on Access Control Models and Technologies, pp. 10-20, May 2001.

[8] E.C. Lupu, D.A. Marriott, M.S. Sloman and N. Yialelis, "A policy based role framework for access control," First ACM/NIST Role Based Access Control Workshop, Gaithersburg, USA, Dec. 1995.

[9] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role based access control models," IEEE Computer, 29(2):38-47, Feb. 1996.

[10] A. A. Pirzada and C. McDonald, "Kerberos Assisted Authentication in Mobile Ad-hoc Networks," in Proc. 27th Australasian Computer Science Conference (ACSC'04), Dunedin, New Zealand, 26(1):41-46, January 2004.

[11] Lalana Kagal, Tim Finin, and Anupam Joshi, "Trust-based security in pervasive computing environments," IEEE Computer, 34(12):154–157, December 2001.

[12] G. Ya-Jun, H. Fan, Z. Ping-Guo and L. Rong, "An Access Control Model for Ubiquitous Computing Application," in Proc. Second International Conference on Mobile Technology, Applications and Systems, 2005.

[13] D. Gambetta, "Can we trust trust?" Trust: Making and Breaking Cooperative Relations, (13):213–237, 2001.

[14] H. Q. Ngo, A. Shehzad, K. A. Pham, M. Riaz, S. Liaquat and S. Y. Lee, "Developing Context-aware Ubiquitous Computing Systems with a Unified Middleware Framework," Embedded and Ubiquitous Computing (EUC 2004), pp. 672-681, Aug. 2004.

[15] Riaz Ahmed Shaikh et al., "Intrusion Tolerant Group-based Trust Management Scheme for Wireless Sensor Networks", submitted for publication.