m-LPN: An Approach Towards a Dependable Trust Model for Pervasive Computing Applications

Munirul M. Haque and Sheikh I. Ahamed
Marquette University, Milwaukee, Wisconsin, USA
{ iq, mhaque}@mscs.mu.edu

Abstract

Trust, the fundamental basis of 'cooperation' – one of the most important characteristics for the performance of pervasive ad hoc network-- is under serious threat with the emergence of counterfeiting and malicious activity. Several constraints exist in the pervasive computing environment such as a device's memory, battery power and computational capability. This results is a high degree of dependence on other devices within the network to provide services. Along with this is the lack of a secure communication medium which invites both active and passive eavesdroppers. In order to restrict the participation of malicious devices, we propose a trust model which is actually a modified form of the well known LPN (Learning Parity with Noise) based Hopper – Blum [1, 2] protocol. Our new refined model has been named as m-LPN (Modified Learning Parity with Noise) which is presented in this paper along with an illustrative example.

Keywords: Trust, LPN, Boolean inner product, authentication and MARKS.

1. Introduction

Advancement of modern wireless technology and the availability of low cost portable devices like the PDA, smart phone, etc. have resulted in the rapid growth of pervasive computing. These devices can be connected wirelessly via Bluetooth or 802.11 and thus form a wireless ad hoc network where any device can join and leave arbitrarily. The performance of an ad hoc network is greatly dependent on the trustworthiness and mutual cooperation of the devices; however, a degree of doubt and uncertainty are integral characteristics of such a volatile environment. Again, the issue of trust emerges which is yet to be efficiently resolved.

Pervasive ad hoc networks can be extremely volatile. As a consequence it is not possible to incorporate trustworthiness in a pervasive ad hoc network like several approaches of a wired network., such as the well known public key infrastructure (PKI). Here, we can not fix a device as omnipresent trusted certificate authority [3] in pervasive computing environment. The approach of Sun and Song [4] is based on a distributed algorithm and game theory. At present [5, 6, 7] the trend is to determine the trust value using a distributed approach where the burden of trust calculation of any device has been placed on the shoulder of each device present in the ad hoc network. Recommendations, activity monitoring, and context monitoring are the important terms to be considered in this approach. Some of the required characteristics of a trust model are a tiny memory footprint, customizabable features, ability to handle several malicious attack scenarios, and regular update of trust value [12].

In [1, 2] on LPN (Learning parity with noise) Nicholas J. Hopper and Manuel Blum have devised a protocol for human – computer authentication where the user has to prove his authenticity to a computer through a dumb terminal. In order to withstand the passive hackers, the model incorporates a noise feature or an intentionally incorrect answer. In our proposed model, we discarded the noise injection phase and modified the protocol to perfectly fit in a pervasive ad hoc scenario. At the same time we introduced a set of new terms including 'net reliability value' and 'threshold'.

Several related works in modeling trust have been described in Section 2. Section 3 uses an example to delineate and explain LPN, followed by the definition and characteristics of our proposed m-LPN in Section 4. Section 5 provides a step-by-step approach. The algorithms and data flow diagrams for both the sender and receiver have been portrayed in this section. Section 6 shows diagrams that contain the architecture of an m-LPN authentication mechanism. An illustrative example that describes the working methodology of the model is shown in Section 7. Some open questions and proposed future guidelines are described in Section 8.

2. Related Work

In 1998, Abdul Rahman and Hailes [8] introduced the term 'distributed trust model' for the first time

which uses a quantitative scheme for the calculation of trust. Each device holds a trust value ranging from -1 (completely untrustworthy) to 4 (completely trusted).

Pirzada and McDonald [5] proposed a model based on an 'effort/return' mechanism for calculating trust. This model incorporates a trust agent in every device that accumulates necessary data for determining trust. This distributed model requires constant monitoring and a devaluation of perceived trust for malicious devices.

A decentralized trust model 'PTM' [9] based on a public key has been proposed. This model reduces the intervention of the user using autonomy of the devices without incorporating any central fixed infrastructure which was required in public key certification. Here the initial trust value is formed through prior knowledge which actually builds a 'believe space'. Later, based on the feedback of actions, an 'evidence space' is formed. The range of trust value has been defined from 0 to 1 where 0, 0.5 and 1 represents 'complete distrust', 'low trust' and 'complete trust'.

According to the model of Sun and Song [4], the trust value will be calculated based on the value assigned the device's reputation, and on various context such as the environment or time. A device's reputation is comprised of the testimony of other devices and feedback of previous actions. In this model each device has to broadcast its reputation value (trustworthiness) while entering the network, thus creating a susceptibility to malicious users who can misrepresent their trustworthiness. Another flaw of this model is malevolent devices cannot work in group.

In a recent work [10, 11], the authors have shown that the Hopper-Blum protocol can be used to increase the security feature of RFID where RFID has been taken as a representative of low cost pervasive devices. The Hopper-Blum protocol uses single client-server configuration. But the configuration and characteristics are different in an ad hoc network scenario.

3. LPN Problem

Given a $\mathbf{q} \times \mathbf{n}$ matrix **A**, where **q** is a polynomial of **n** in size, a **q** bit vector **z**, and a noise parameter $\boldsymbol{\eta} \in (0,1/2)$, find an n bit vector x such that $|\mathbf{Ax-z}| \leq \boldsymbol{\eta}\mathbf{q}$. [1, 2]

3.1 Explanation of LPN

Let us consider a scenario where **A** and **B** share a common $\boldsymbol{n}$ bit secret $\boldsymbol{x}$. First, **A** sends an arbitrary challenge $\boldsymbol{a} \in \{\mathbf{0,1}\}^{\mathbf{n}}$ to **B**. Both **A** and **B** calculate $\boldsymbol{a.x}$ modulo 2 which actually denotes the parity bit $\boldsymbol{z}$. **A** will receive this parity bit from **B** and match it with its own calculated result. **B** will be accepted if $\boldsymbol{z}$ matches with the parity bit calculated by **A**.

Since the parity bit can be only 0 or 1, a passive snooper can guess the parity bit with probability $\mathbf{2^{-1}}$. If **A** sends $\boldsymbol{q}$ challenges, then the possibility of making a correct guess in all the $\boldsymbol{q}$ rounds is $\mathbf{2^{-q}}$ which is very small for a large value of $\boldsymbol{q}$. But if a passive eavesdropper observes minimum $\boldsymbol{n}$ challenge – response pairs, the eavesdropper can regenerate the secret $\boldsymbol{x}$ through the Gaussian elimination method.

In order to handle this scenario, a term $\boldsymbol{\eta}$ named noise has been introduced. **B** can intentionally choose an incorrect answer with probability $\boldsymbol{\eta}$. **B** will be considered a valid device if the number of correct answers is greater than or equal to $\mathbf{(1- \eta) \times q}$.

4. m-LPN

We define our proposed m-LPN as follows: Given a $\mathbf{q} \times \mathbf{n}$ matrix **A**, where **q** is a polynomial of **n** in size, a **q** bit vector **z**, and a threshold parameter $\boldsymbol{\Omega} \in (0,1)$, authenticate the incoming device if Reliability_Value_Calculate$(\mathbf{Ax , z}) \geq \boldsymbol{\Omega}$.

4.1 Characteristics of the m-LPN

1. The LPN problem is based on a single server-client scenario. But in an ad hoc network there is no static device that can function as a server. Rather than burdening a specific node with the responsibility of sending all challenges, we have distributed this responsibility to all nodes present in the network. Every node in the network sends one challenge to a node that wants to join the network.

2. A node has been termed as 'leader node' which has been chosen on the basis of trust level and battery power. Leader node is responsible for generating new secrets and summing up 'partial reliability values' to provide a specific trust value for the new node. The term 'partial reliability value' is defined below.

3. Here, we have omitted the introduction of a noise feature. We can choose the length of the secret $\boldsymbol{x}$ in such a way that outweighs the number of devices present in the network. For example if the highest number of devices in the ad hoc network is $\boldsymbol{i}$ and if the length of the secret $\boldsymbol{x}$ is anything greater than or equal to $\boldsymbol{i+1}$, then the passive intruder will not be able to regenerate $\boldsymbol{x}$ by capturing sample challenge-response pairs. As a result the purpose of noise introduction becomes invalid. Let us consider a scenario of 5 nodes. Assume that the length of the secret is 6. If a new node arrives and provides correct answers to all

challenges, it then will be permitted to join in the network. The next time another device tries to join the network, the number of challenge – response pairs generated will be 6 which equals the length of the secret. In order to ensure that the length of the secret will always be greater than the number of challenge – response pairs generated, the leader has been given the authority to dynamically change the length of the secret. The leader will generate an arbitrary secret of the required length, encrypt it and send it to other nodes. Other nodes will decrypt the message and get the new secret whose length ensures that a passive

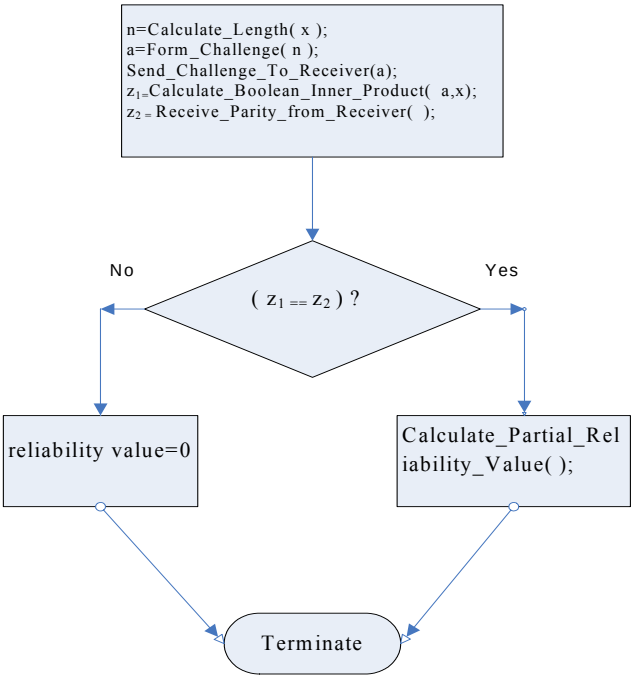

Figure 1: Data flow diagram of the sender

snooper will not be able to regenerate the secret by just observing challenge – response pairs. Whenever a node joins or leaves, it broadcasts this message in the network. As a result the leader node always knows the exact number of devices present in the network. After a new node joins the network, the leader node generates a new secret with proper length (if the present number of node is ***i+1*** the length of the new secret will be ***i+2***) and sends it to all the present nodes and to the authentication mechanism. As a result when a new node comes through the authentication mechanism and tries to join the network, it knows the current secret.

4. We have used the term 'net reliability value' as a synonym of trust in this paper. Net reliability value, which is actually very much self-explanatory, represents the reliability of that device in the network. As the invited device is achieving a specific partial reliability value at each challenge-response round, we can use several reasoning techniques to identify the net reliability value for the invited device. The reasoning technique can range from a simple average function to probability or fuzzy logic. When a new device gives a correct answer to a challenge, the challenge sender recommends its own trust level as the reliability value for the new node to the leader node. This recommended value is known as 'partial reliability value'. A new node will receive a 'partial reliability value' from each node present in the network.

5. We have introduced the term 'threshold' in the authentication procedure which indicates the minimum reliability value required to join the network. Threshold encapsulates one of the major characteristics of pervasive computing termed 'context awareness'. The threshold value is customizable based on the contextual information at a specific moment, thus permitting context to play a vital role in the trust mechanism.

5. Details of m-LPN

Here we assume that each device that has passed the authentication mechanism and is now trying to join the network knows a specific secret ***x***. If any malicious device bypasses the authentication phase and takes part in the challenge – response phase in order to join in the network, it will not have any prior knowledge about the secret ***x***.

1. Let us assume that there are ***m*** devices $x_1, x_2, x_3, \ldots, x_m$ in the network. All the devices in the network share a random ***n*** bit secret ***x.***

2. When a device (let us assume it is '***y***') wants to join the ad hoc network, it has to receive an invitation from at least one member device in the network.
3. Each device in the network will send a random challenge $\{0,1\}^n$, for example ***a,*** to ***y***.

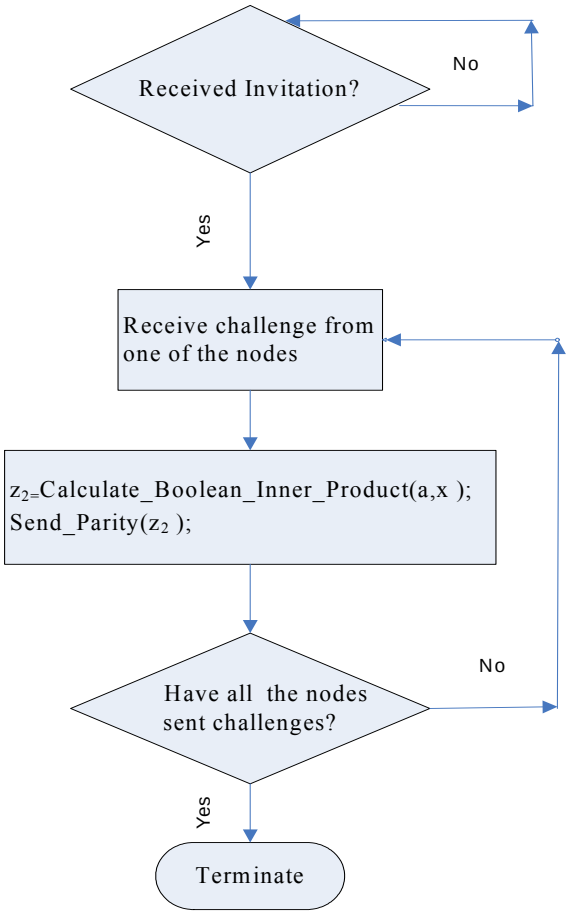

Figure 2: Data flow diagram of the receiver

4. Then both the sender and receiver device will calculate the Boolean inner product ***a.x*** to denote the parity bit. We are considering modulo 2 arithmetic here.
5. The sender will check its own parity bit after getting parity response from ***y.*** If both parity bits match, then ***y*** will get a specific point based on the reliability value of the sender. Sender will submit this 'partial reliability value' to the leader node.
6. This process will be repeated for each of the devices present in the network.
7. Finally, using a selected reasoning technique, a net reliability value for ***y*** will be generated by the leader.
8. If this net reliability value passes a pre-specified threshold value (this value will actually depend on several contexts such as situation, time, sophisticated level of the data contained by the member devices, etc.), the new device will be permitted to join the network.

Data flow diagram of sender and receiver are given in Figure 1 and Figure 2 respectively.

6. Architecture

MARKS [13, 14] middleware provides the core communication facilities along with other services such as Knowledge usability [15], Resource discovery [16] and PerAd service [17]. m-LPN is being added as a middleware service to MARKS. The placement of m-LPN authentication service has been shown in the following Figure 3.

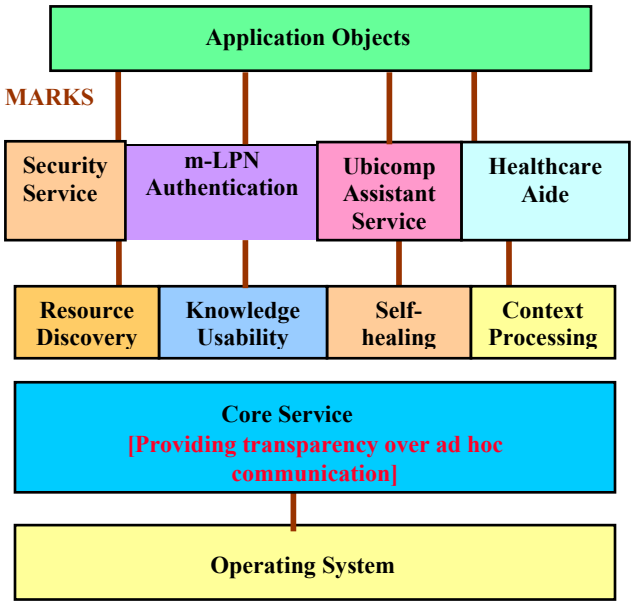

Figure 3: MARKS architecture [13, 14]

The several components of m-LPN authentication architecture (Figure 4) have been depicted below:

Receiver and Broadcaster	**Random challenge generator**	**AND operation performer**
XOR operation performer	**Reliability value calculator**	**Threshold value calculator**

Figure 4: m-LPN Authentication Architecture

Receiver and Broadcaster component is responsible for all types of send and receive operations. **Random challenge generator** forms the arbitrary ***n*** bit challenge ***a***. **AND** and **XOR operation performer** are needed to carry out the operations of bit wise AND between secret ***x*** and challenge ***a*** and XOR of all the bitwise ANDs. They actually implement the Boolean inner product **(*a.x*) modulo 2**. **Reliability value calculator** will get the parity bits calculated by the sender and receiver as input from which it will calculate the partial reliability value. This partial reliability value will be sent to the leader. Then using the selected reasoning technique the net reliability value will be generated. **Threshold value calculator** will compute the current minimum reliability value which is required by a device to join the network. This value will be changed dynamically with the passage of time depending on several contexts.

7. An Illustrative Example

Mrs. Berry had been suffering from some critical gynecological problems since last week. When she came to 'Mount Aurora' hospital she was admitted under Dr. Masson, and underwent some diagnostic procedures. Dr. Masson received the results and stored the information in a file in his PDA. Mrs. Berry now wishes to look at that information. It is possible for Mrs. Berry's mobile device to access this information, but first her device needs to join the ad hoc network whose present members are Dr. Masson, Dr. Morrice, Dr. Carl and Mr. Cary - the administrator. We will call these devices as S1, S2, S3 and S4, respectively, and that of Mrs. Berry will be R. At the beginning step, S1 sends the invitation to R because S1 knows R. Let us assume that the secret ***x*** is 111001. As we know, in order to restrict any passive intruder from recalculating the secret ***x*** we need the length of ***x*** to be greater than the number of devices present in the network. We have chosen the length of the secret ***x*** to be 6. At first S1 will generate an arbitrary challenge 011000 and send it to R. S1 will now calculate the Boolean inner product ***a.x* modulo 2,** which is actually bit wise AND operation between 011000 and 111001 and then the XOR of all the bitwise ANDs. The result of bit wise AND will be 011000 and the XOR result is 0 ($0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0$). Considering even parity, this result also denotes the parity bit. R will also calculate the parity bit in the above fashion and send it to S1. If both the parity bit matches, then the reliability value of S1 will be sent to the leader as the partial reliability value of R from S1. If they don't match, a partial reliability value of 0 will be sent to the leader. This process will be continued once for each of the four devices. The following table has summarized the results:

Table 1: Summary of authentication result

Sender	Reliability /trust value of sender	**a**	**a.x modulo 2**	Z_1	Z_2	Partial reliability value
S1	.67	011000	0	0	0	.67
S2	.70	101010	0	0	0	.70
S3	.89	010101	0	0	1	0.0
S4	.60	000111	1	1	1	.60

In the above table the reliability/trust value of the senders, arbitrary challenges ***a,*** and the parity bit returned by R at different rounds (z_2) have been assumed. Assuming that R sends a wrong parity bit in round 3, we have placed a partial reliability value of 0.

Based on the reasoning technique adopted, a net reliability value for R will be calculated. If this value is greater than the present threshold value, then R will be permitted to join the network.

After joining, R (that is Mrs. Berry) can get the file from Dr. Masson's device (S1) if the required reliability level for file transfer in S1 is smaller than or equal to the reliability level of R.

If a malicious device tries to join the network it will be able to answer correctly each challenge with a probability of .5 which indicates that it will be able to answer 50% challenges correctly. As a result this malicious device will receive 0 as a partial reliability value for 50% of the time. This will ensure a poor trust value and guarantee that the malicious device will not be able to join the network.

8. Conclusions and Future Works

In this paper, we adopted the well known Hopper-Blum algorithm in an ad hoc scenario and presented several pros and cons about a methodology, m-LPN, for possible adaptation. We implemented the core services of MARKS [13-14] middleware and some other services [18-19]. Currently, we are implementing m-LPN using VC# and .Net Compact Framework on Dell Axim X50v PDAs. We will add this m-LPN as an authentication service to MARKS. We will measure the impact of implementing m-LPN as the authentication service on other crucial metrics like power consumption or signal strength. We will evaluate the performance such as scalability of m-

LPN authentication service on a simulated large ad hoc network using OMNet++.

There are several open questions in this research area. Some example questions can be as follows:

1. Should each of the devices throw a challenge or a selected number of devices? Or should a specific group of devices, selected based on some specific criterion, get this opportunity?
2. Among several reasoning techniques, such as simple average, Bayesian probability, and fuzzy logic, what type of reasoning technique should be most suitable and appropriate?
3. What should be the criterions to consider changing the threshold value dynamically?

In the future, we will address these issues and try to provide an authentic trust model within the several constraints of pervasive devices.

Acknowledgement

The authors appreciate the help of Paula Stroud for assisting us with valuable comments and reviews.

References

[1] N. Hopper and M. Blum. A Secure Human-Computer Authentication Scheme. Technical Report CMU-CS-00-139, Carnegie Mellon University, 2000.

[2] N. J. Hopper and M. Blum. Secure Human Identification Protocols. In *Advances in Cryptology - ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 52–66, 2001.

[3] J. Weise, "Public Key Infrastructure Overview," Sun Blueprints Online, Aug. 2001, http://www.sun.com/blueprints/0801/publickey.pdf

[4] H. Sun and J. Song, "Strategyproof Trust Management in Wireless Ad Hoc Network", *Proceedings of the IEEE Canadian Conference on Computer and Electrical Engineering*, 2004.

[5] S. Yi and R. Kravets. "Key Management for Heterogeneous Ad Hoc Wireless Networks," *Proceedings of the 10th IEEE International Conference on Network Protocols*, pp. 202-203, 2002.

[6] A. Pirzada and C. McDonald, "Establishing Trust in Pure Ad-hoc Networks," *Proceedings of the 27th conference on Australasian computer science*, vol. 26, 2004.

[7] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang, "Self-securing Ad Hoc Wireless Networks," *Proceedings of the Seventh International Symposium on Computers and Communications (ISCC'02)*, pp. 567, 2002.

[8] A. Abdul-Rahman and S. Hailes, "A Distributed Trust Model," *Proceedings of the 1997 workshop on New security paradigms*, 1998.

[9] F. Almenarez, A. Marin, C. Campo, and C. Garcia, "PTM: A Pervasive Trust Management Model for dynamic open environments," *Pervasive Security, Privacy, and Trust (pspt 2004), Massachusetts, 2004*.

[10] S.A. Weis, "Security parallels between people and pervasive devices," *Pervasive Computing and Communications Workshops, 2005. Third IEEE International Conference ,2005* Page(s):105 – 109.

[11] A. Juels and S. A. Weis." Authenticating Pervasive Devices with Human Protocols." *In Submission*, 2005.

[12] S. T. Wolfe, S. I. Ahamed, and M. Zulkernine, "A Trust Framework for Pervasive Computing Environments", to appear in *The 4th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-06)*, IEEE CS Press, Dubai, UAE, March 2006.

[13] MARKS: a middleware for pervasive computing of Ubicomp Research Lab. (www.mscs.mu.edu/~ubicomp).

[14] M. Sharmin, S. Ahmed, and S. I. Ahamed, "MARKS (Middleware Adaptability for Resource Discovery, Knowledge Usability and Self-healing) for Mobile Devices of Pervasive Computing Environments, " To appear in the *Third International Conference on Information Technology : New Generations (ITNG 2006),* April, 2006, Las Vegas, Nevada, USA.

[15] S. Ahmed, M. Sharmin, and Sheikh I. Ahamed, "Knowledge Usability and Its Characteristics for Pervasive Computing," *The 2005 International Conference on Pervasive Systems and Computing(PSC-05),* Las Vegas, NV, USA, June 2005, pp. 206-209.

[16] M. Sharmin, S. Ahmed, and S. I. Ahamed, "SAFE-RD (Secure, Adaptive, Fault Tolerant, and Efficient Resource Discovery) in Pervasive Computing Environments", *IEEE international Conference on Information Technology (ITCC 2005),* Las Vegas, NV, USA, April 2005, pp. 271-276.

[17] S. Ahmed, M. Sharmin, and S. I. Ahamed, "PerAd-Service: A Middleware Service for Pervasive Advertisement in M-Business," *The 29th Annual International Computer Software and Applications Conference (COMPSAC 2005), IEEE CS Press, vol 2,* Edinburgh, Scotland, July, 2005, pp. 17-18.

[18] M. Sharmin, S. Ahmed, and S. I. Ahamed, "Ubicomp Assistant: An Omnipresent Customizable Service using MARKS," *Proceedings of the 21st Annual ACM Symposium*

on Applied Computing (ACM SAC 2006), Dijon , France, April, 2006, pp. 1013-1017.

[19] M. Sharmin, S. Ahmed, and S. I. Ahamed, "An Adaptive Lightweight Trust Reliant Secure Resource Discovery for Pervasive Computing Environments, " " *Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computer and Communications (PerCom 2006)*, Pisa, Italy, Mar 2006, pp. 258-263.