Sixth Annual IEEE International Conference on Pervasive Computing and Communications

A Risk-aware Trust Based Secure Resource Discovery (RTSRD) Model for Pervasive Computing

Sheikh I. Ahamed[1], Moushumi Sharmin[2], and Shameem Ahmed[2]
[1]*Marquette University, Milwaukee, Wisconsin, USA*
[2]*University of Illinois at Urbana-Champaign, IL, USA*
iq@mscs.mu.edu, ashameem38@gmail.com, nisha_moushumi@yahoo.com

Abstract

To address the challenges posed by device capacity and capability, and also the nature of ad-hoc network of pervasive computing, a resource discovery model is needed that can resolve security and privacy issues with simple solutions. The use of complex algorithms and powerful fixed infrastructure is infeasible due to the volatile nature of pervasive environment and tiny pervasive devices. In this paper, we present a risk-aware trust based secure resource discovery model, RTSRD (Risk-aware Trust Based Secure Resource Discovery) for a truly pervasive environment. Our model is an adaptive hybrid one that allows both secure and non-secure discovery of services on adaptive trust. RTSRD also incorporates a risk model for sharing resources with unknown devices. Hence, the two contributions of this paper are: adaptive trust, and risk model for resource discovery in pervasive computing environments.

Keywords: MARKS, Resource Discovery, Risk model, Secure Service and Device discovery.

1. Introduction

Pervasive computing [1], [2], [3] has evolved over the last few years due to recent developments in portable low-cost lightweight devices and the emergence of short range, and low power wireless communication networks. In a pervasive computing environment, there are different kinds of networks. On one end, some tiny devices communicate among themselves with the support of fixed, powerful devices. These devices act as servers or proxies and handle complex computations on behalf of the tiny devices. On the other end, some devices form an ad hoc network. In this environment, there is no fixed infrastructure support. The devices communicate with each other directly or via another mobile device, and are responsible for performing computations by themselves. Despite the exponential growth of the exploitation of handheld devices (e.g. PDAs, laptops, smart phones etc.), these devices themselves are suffering from a number of limitations [7], [8], which include but is not limited to, inadequate processing capability, restricted battery life, limited memory space, slow expensive connections, frequent line disconnection, and confined host bandwidth. Our focus is on this type infrastructure-less pervasive computing area, which leads to the dependency on other devices for resources. The nature of devices, communication pattern, and dependency on others in turn causes security threats. Also, due to the ad hoc and ephemeral nature of the network, one can't expect to get service from a particular device for a long span of time. Hence, resource discovery is an integral part of every system running in a pervasive computing environment [6].

The significance of security during resource discovery in pervasive computing environments is an established truth [9], [10], [11]. Privacy, security, and trust issues in resource discovery in pervasive computing area are of utmost importance [4]. Many users may be happy to share the resources of their handheld devices, provided this sharing will not cause any security threat to them. Thus, the resource discovery process demands models that ensure the privacy and security of the user. The traditional security mechanism does not work in this environment, because the devices are computationally poor and the notion of physical security is not applicable [12]. Lack of availability of information about users is another primary concern in designing a discovery model, which necessitates the introduction of risk assessment [13]. Existing resource discovery models can be divided into three broad categories. First are the resource discovery models that do not address security issues [14], [15], [16], [17], [18]. Second, there are models that consider a full-fledged security mechanism with the help of some fixed infrastructure support (powerful servers, proxies, etc.) [19], [20], [21]. There are also models that support security with the assistance of additional hardware [22], mutual authentication [23], and trust [24], [25]. There are few models [32-22], which can be used in ad hoc environment but none of them take risk into account.. These models either completely trust or completely distrust one device. Each of these models has their own strength and weaknesses as they attempt to solve the problem using different approaches.

In this paper, we present a risk-aware trust based secure resource discovery model, RTSRD (Trust Based Secure Resource Discovery). Our model is designed for a truly pervasive environment, where we assume that the mobile devices would be able to handle necessary

0-7695-3113-X/08 $25.00 © 2008 IEEE
DOI 10.1109/PERCOM.2008.113

Authorized licensed use limited to: Marquette University. Downloaded on December 7, 2009 at 15:03 from IEEE Xplore. Restrictions apply.

computations and communications by themselves, without any fixed infrastructure support. Our model is a hybrid one, that allows both secure and non-secure discovery of services. This model allows resource discovery and sharing based on mutual adaptive trust. However, for unknown devices building trust relationships is complicated and sometimes impossible. To handle situations like this, we also include a risk model.

The outline of this paper is as follows: An overview of our proposed approach is illustrated in section 2. The details of the models have been described in section 3 and 4. The evaluation of our proposed model is presented in section 5. Our future research direction and concluding remarks are described in section 6.

2. Overview of RTSRD

To address the challenges presented in the introduction section, we propose a secure resource discovery model, RTSRD (Risk-aware Trust Based Secure Resource Discovery). The RTSRD model contains a discovery model and a trust, risk, and security management unit. The RTSRD model consists of SAFE-RD (Secure, Adaptive, Fault Tolerant, and Efficient Resource Discovery)[35] and SSRD (Simple and Secure Resource Discovery) sub units. The SAFE-RD contains the device discovery unit and the resource discovery agent while the SSRD unit contains the trust, risk, and security unit. Initial result on Trust of SSRD was presented in [36] and risk was introduced in [37]. The details of the SAFE-RD model and the MARKS architecture have been published in [35] and [26] respectively. This paper addresses adaptive trust, risk and security.

The SSRD unit handles security related issues and consists of *trust management, risk assessment,* and *security management* sub units. The SSRD unit is directly linked to the resource discovery agent. The functionalities of all these units are maintained and controlled by the resource manager. All these units provide users with privacy and security without explicit user interaction. The model requires initial user input to set security levels for different services provided by the device. After this point, it needs user permission only in case of a highly secure service sharing time. This is necessary to maintain users' privacy.

The trust management unit is responsible for maintaining the trust relationship with other devices. This unit calculates trust values for all devices and also updates the trust values depending on the behavior of the service provider or requester. It maintains a list of service-specific and average trust values and communicates with the risk assessment and security management unit whenever necessary.

For secure discovery and communication among devices, we are using a trust model influenced by PGP (Pretty Good Privacy) [34]. However, Direct PGP is not used since it requires huge computation. We are considering not only average trust but also service specific trust that will make the model more robust. For trust calculation, we are considering previous interaction information and also recommendation from other neighboring devices.

Lack of historical information is a natural phenomenon in a truly pervasive environment. Even if there is no information available, we still want to receive and provide services. To handle situations like this, we are proposing a risk assessment model that allows us to share resources without compromising the security of the device

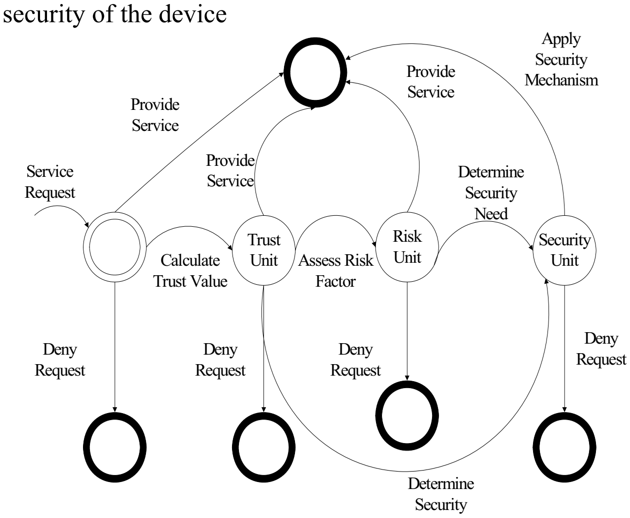


Fig. 1. Conceptual Diagram of RTSRD Model

The *security management unit* selects the mode of communication (broadcast, multicast or unicast) depending on situation and security needs for a specific service. It also facilitates secure sharing of services. This mode is selected depending on the predefined functions and does not require explicit user intervention each time a service is requested. The security model also determines the mode of communication. It works with the trust management unit to request service and to provide service as securely as possible without compromising the performance of the device. Fig. 1 shows the conceptual diagram of the RTSRD model.

3. Adaptive Trust Model

Our trust model is introduced in [36] and enhanced in this paper. In our model, the resource manager of each device contains a list of nearby devices, which discloses their presence along with an associated trust value. Equation 1 is used for the calculation of trust values for devices that have interacted with this device earlier.

Authorized licensed use limited to: Marquette University. Downloaded on December 7, 2009 at 15:03 from IEEE Xplore. Restrictions apply.

$$\tau(SP, A) = (\sum_{i=1}^{n} S_i * \tau(SP_i, A, x)) / \sum_{i=1}^{n} S_i$$
……. 1

Here, SP = Service Provider
τ(SP, A) = Average Trust value of device A for device SP
Si = Security level of i-th service
τ(SPi, A, x) = Trust value of A for i-th service
n = Number of services that links SP and device A

In case a new device joins the pervasive network, equation 2 is used by resource manager to calculate the trust value.

$$\tau(SP, D_{new}) = (\sum_{i=1}^{n} \tau(SP, i) * \tau(i, D_{new})) / \sum_{i=1}^{n} \tau(i, D_{new})$$
… …………………………. 2

Here, SP = Service Provider,
Dnew = New device requesting service
τ (SP, Dnew) = Average Trust value of Dnew for device SP
τ (i, Dnew) = Average Trust value of device i for Dnew
n = Number of services that links SP and A

A human-like trust relationship is introduced in [37] and the average trust level value can be updated using following equations *3, 4*, and *5* but the details of c were not modeled.

$$\theta_i = (\sigma_a - \sigma_r) / \sigma_a \quad \text{… ……………………… 3}$$

$$\tau(SP, D) = \tau(SP, D) + \theta_i \quad \text{… ………… ……. 4}$$

$$\tau = \frac{\sum_{i=1}^{n} \theta_i * \omega_i}{n} \pm c \quad \text{… …………………………. 5}$$

Here, *θ* = Modification value for service *i*,
τ(SP, D) =Trust value of device *D* for service SP,
τ(D = Average trust value of device *D*,
ω = 0.1* security level of service *i*,
n = Number of services relating to provider and requester,
σ = Required time for a successful request completion,
σ_a = Average service time, and
c = Random behavioral parameter.

In this paper, c is modeled, which is a major contribution of this paper. In *equation 5*, *c* is included to reflect behaviors such as sending too many requests in a small amount of time, repeatedly sending a request that has already been rejected, and so on. *c* is generated from a uniform random number generator using parameters like number of requests (μ), number of accepts (α), number of rejects (β), number of same request (ω), etc. If *c*>0.5, then we add in *equation 4*, otherwise we deduct *c*. The *c* value is calculated using *equation 5*

$$c = \left(\frac{\alpha}{\beta}\right) - \left(\frac{\omega}{\mu}\right) \quad \text{… ………………………. 6}$$

Our trust model is privacy aware. It consults the user when responding to a higher security level service request. Even if the trust or risk model indicates that it is safe to share a service, the user can deny any request. The model is designed in a way that it does not prompt the user for permission for every service sharing request. However, when the security level of a particular service is above a threshold value, it asks for user permission. Thus, it maintains transparency of operation and protects users' privacy.

4. Risk Model

A risk model is essential during the sharing services in a pervasive environment. Risk evaluation becomes significant when a service request comes from an unknown device or when there is not enough recommendation information. When a service request arrives, we calculate the trust value of the requesting device (if the providing device has information about the requester or by collecting recommendation from other devices). Then based on the security level of the requested service, we accept or deny the request. When the requester is unknown to all the neighboring devices (a very common scenario in pervasive computing), the device is assigned an initial trust value of 0.5 which would allow it to receive lower security-intensive services and build a trust relationship with others. However, if that device requires a higher security level service, it is denied. To address this issue, we have added the risk assessment along with our trust and security model.

The risk model that we are currently using is a lightweight one. Each device has a risk evaluator. This evaluator stores information about high security services and calculates the risk value when a request comes for one of these services. Each time a service request arrives along with an accepted or rejected event, it updates the risk value associated with that service. It collects information about the service that includes number of accepts (γ), total number of requests (ϕ), average trust values of the devices who request this service, service time (σ), etc.

Authorized licensed use limited to: Marquette University. Downloaded on December 7, 2009 at 15:03 from IEEE Xplore. Restrictions apply.

Table 1: Risk calculation

Id	Number of Request (ϕ)	Number of Accept (γ)	Average Trust Value (τ)	Average Service Time (σ) in ms
6	4	1	0.72	20
10	6	4	0.6	15
14	15	12	0.85	35
…	…	…	…	…

Table 2: Service discovery

Service Name	Time (Sec)		
	Normal	Trust	Trust, Risk, & Security
DateTime	0.1	0.105	0.11
WAV (148KB)	0.7	0.72	0.8
Chat SW (262KB)	0.9	0.925	0.98
Unzip SW (323 KB)	1.0	1.03	1.1
Address book (810KB)	1.8	1.91	2.1
Dictionary (5.94MB)	17.2	17.25	18.1
Music SW (7.96MB)	23.6	23.66	23.7
Acrobat Reader (13.5 MB)	40	40.05	40.1

To calculate the risk factor (7) is used-

$$\rho = \left(\frac{\gamma}{\phi}\right) \cdot \tau$$

(7)

Here,

ρ = Risk Factor

γ = Number of accepts

ϕ = Number of request

τ = Average trust value for this service

The range of the risk factor, ρ is $0 <= \rho <= 1.0$. This is a weighted average with respect to average trust value. A value of 0.5 indicates around 50% acceptance rate for this particular service. If the risk factor value is high (>0.5), then the request is rejected. In the case of a low risk factor, the service is provided. Based on this value, the device assigns a risk factor with the service. As this information is collected every time a service is requested or shared, a historical database is created for services of a particular device. Each device has its own database that allows it to decide the risk factor for its services. This allows a device to decide whether to accept a request or not when there is little or no information available about a requester. Table 2 shows some sample data stored in a device.

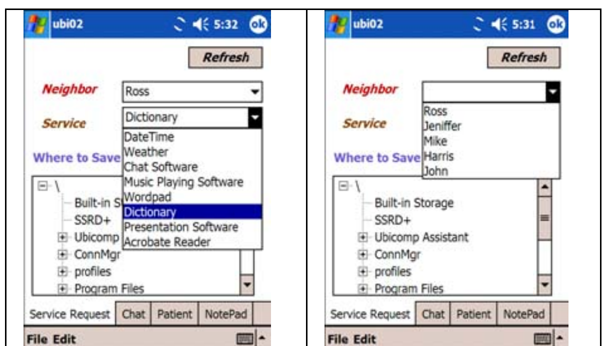
Each time a service request is made, the risk value table is updated to include the modified number of requests, number of accepts, average trust value of devices for which the request is accepted, and average service time to offer. The updated data is used to calculate the risk factor for sharing a service with unknown devices. We are currently using statistical distributions to find out optimal percentage rate and trust value pair that lowers the risk of service sharing. The average service time is compared with the service-sharing time to evaluate the behavior of the requesting device. This value is used for dynamic modification of trust value.

5. Evaluation

We have evaluated the performance and usability of the RTSRD model by implementing prototype. We have designed applications that use the RTSRD model for device discovery and resource sharing. To estimate the overhead of using this model, we have measured the battery power consumption.

5.1. Prototype Implementation

We have implemented a prototype of our proposed model in the resource discovery unit of MARKS. We have used a test bed consisting of a set of Dell Axim X30 pocket PCs (Processor type is Intel@PXA270, speed is 624 MHz). The underlying OS is WinCE and the implementation language is C# on .NET Compact framework. This prototype is also compatible to laptops, desktops, and smart phones. As the underlying wireless protocol, we have used the mobile ad hoc mode of IEEE 802.11b. Some screenshots of application using the resource discovery service are shown in Fig. 2



(a) (b)

Fig. 2. Application That Uses RTSRD (a) Available services. (b) Neighboring devices.

5.2. Performance Measurement

Our resource discovery model is lightweight. To evaluate the performance of our model we have used battery power as a performance metric. We have constructed a test bed of seven PDAs, which are wirelessly connected in ad-hoc mode. At first, we have measured the

Authorized licensed use limited to: Marquette University. Downloaded on December 7, 2009 at 15:03 from IEEE Xplore. Restrictions apply.

power without running our prototype. Later we have done the same thing after executing the prototype. Fig. 3 shows the remaining battery power for seven PDAs before and after running RTSRD model. It shows that the battery power consumption is nominal for TSRD.

To collect data for comparisons, we generated random service requests from seven devices. We measured the time required for service discovery and sharing using our model and without using our model (normal case). A portion of the collected data is shown in Table 2.

Here, we see that for normal services (e.g. DateTime, Chat & Music playing SW, etc.), encryption and user intervention is not needed. To calculate the trust value, it needs less than 60 ms. On the contrary, for the delicate address book sharing, both user intervention and encryption are needed. The encryption and trust calculation take only 110 ms, which is negligible.

We have collected data of our model using only the trust model and using trust, risk, and security model. We have compared both sets of data to estimate the overhead of using the risk and the security models.

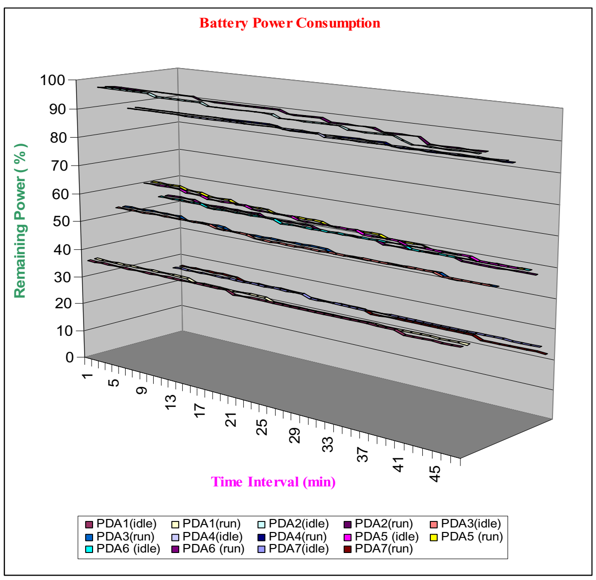


Fig. 3. Power Consumption by RTSRD

6. Conclusion and Future Work

In this paper**,** we have proposed risk-aware a resource discovery model, RTSRD. To maintain the privacy of users and their willingness to share resources, trust, and risk models have been implemented. Our model is a hybrid one in a sense that it operates both in secure and non-secure mode depending on the level of security needs for the service. By implementing a hybrid mode of operation, we have minimized the overhead of encrypting messages each time a device requests or provides services. However, when there is no prior information available, building a trust relationship is difficult. To address situations like this, we have also added a risk model that analyzes the risk of sharing a particular resource and takes appropriate action. The addition of appropriate risk parameters will make this model tremendously useful. We have implemented RTSRD as a part of MARKS, a dependable middleware designed for devices running on a pervasive computing environment. We have also implemented applications that use RTSRD.

Our existing model works for single-hop resource discovery and sharing. This model can be extended to facilitate multi-hop discovery and service sharing. Features like dynamic resource integration can be included as future work.

References

[1] M. Weiser, "Some Computer Science Problems in Ubiquitous Computing," *Communications of the ACM*, Vol. 36, No. 7, July 1993, pp. 75-84.

[2] Pervasive Computing definition, URL: http://www.parliament.vic.gov.au/sarc/EDemocracy/Final_Report/Glossary.htm

[3] Pervasive Computing framework, URL: http://framework.v2.nl/archive/archive/node/text/default.xslt/nodenr-156647

[4] P. Robinson, H. Vogt, and W. Wagealla, "Some Research challenges in pervasive computing," *Post workshop at the second international conference on pervasive computing*, April 18-23, 2004, Vienna, Austria, pp. 1-16.

[5] M. Weiser, "The Computer for the Twenty-First Century," *Scientific American,* September 1991, pp. 94-104.

[6] T. Kindberg and A. Fox, "System Software for Ubiquitous Computing," *IEEE Pervasive Computing*, Jan-Mar, 2002, pp. 70-81,

[7] M. Satyanarayanan, "Fundamental Challenges in Mobile Computing," *Fifteenth ACM Symposium on Principles of Distributed Computing*, Philadelphia, Pennsylvania, USA, May 1996, pp. 1-7.

[8] R. Want and T. Pering, "System challenges for ubiquitous & pervasive computing," *27th International Conference on Software Engineering (ICSE 2005)*, St. Louis, Missouri, USA, May 15-21, pp. 9-14.

[9] F. Stajano, "Security for Ubiquitous Computing," *Wiley*, February 2002, pp. 110-111.

[10] F. Stajano and R. Anderson, "The Resurrecting Duckling: security issues for ubiquitous computing," *Computer*, Vol. 35(4), Part Supplement, April 2002, pp. 22–26.

[11] K. Matsumiya, S. Tamaru, G. Suzuki, J. Nakazawa, K. Takashio, H. Tokuda, "Improving security for ubiquitous campus applications," *Symposium on Applications and the Internet Workshops (SAINT 2004)*, January 2004, pp. 417–422.

Authorized licensed use limited to: Marquette University. Downloaded on December 7, 2009 at 15:03 from IEEE Xplore. Restrictions apply.

[12] L. Kagal, T. Finin, and A. Joshi, “Moving from Security to Distributed Trust in Ubiquitous Computing Environments,” *IEEE Computer*, December 2001.

[13] Y. Chen, C. D. Jensen, E. Gray, V. Cahill, and J. Seigneur, “A general risk assessment of security in pervasive computing,” URL: https://www.cs.tcd.ie/ publications/tech-reports/ reports.03/TCD-CS-2003-45.pdf.

[14] M. Nidd, “Service Discovery in DEAPspace,” *IEEE Personal Communications*, August 2001, pp. 39-45.

[15] B. A. Miller, T. Nixon, C. Tai, and M. D. Wood, “Home Networking with Universal Plug and Play”, *IEEE Communications Magazine*, December 2001, Vol. 39, Issue 12, pp. 104-109.

[16] W. Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley, “The design and implementation of an intentional naming system,” *17th ACM Symposium on Operating Systems Principles (SOSP ’99)*, Kiawah Island, Scotland, 1999, pp. 186-201.

[17] M. Balazinska, H. Balakrishnan, and D. Karger, “INS/Twine: A Scalable Peer-to-Peer Architecture for Intentional Resource Discovery,” *International Conference on Pervasive Computing*, Zurich, Switzerland, August 26-28, 2002, pp. 195-210.

[18] Microsoft Corporation, “Universal Plug and Play Device Architecture,” Version 1.0, Microsoft Co., 2000.

[19] S. Czerwinski, B. Y. Zhao, T. Hodes, A. Joseph, and R. Katz, “An Architecture for a Secure Service Discovery Service,” *Fifth Annual International Conference on Mobile Computing and Networks (MobiCom '99)*, Seattle, WA, 1999, pp. 24-35,

[20] F. Zhu, M. Mutka, and L. Ni, “Splendor: A secure, private, and location-aware service discovery protocol supporting mobile services,” *Pervasive Computing and Communications, 2003 (PerCom 2003),* Proceedings of the First IEEE International Conference, 23-26 March 2003, pp. 235–242.

[21] F. Zhu, M. Mutka, and L. Ni, “PrudentExposure: A Private and User-centric Service Discovery Protocol,” *Proceedings of the 2004 IEEE Annual Conference on Pervasive Computing and Communications (PerCom 2004),* March 2004, pp. 329-340.

[22] H. Kopp, U. Lucke, and D. Tavangarian, “Security architecture for service-based mobile environment,” *Third IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'05),* Washington DC, USA, March 2005, pp. 199-203.

[23] F. Zhu, M. Mutka, and L. Ni, “Expose or not? A progressive exposure approach for service discovery in pervasive computing environments,” *Third IEEE International Conference on Pervasive Computing and Communications (PerCom 2005)*, March 2005, pp. 225–234.

[24] F. Almenarez and C. Campo, “SPDP: A Secure Service Discovery protocol for Ad-hoc networks,” *9th open European summer school and IFIP workshop on next generation networks (EUNICE 2003),* Hungary, September 2003.

[25] R. He, J. Niu, M. Yuan, and J. Hu, “A novel cloud-based trust model for pervasive computing,” *The Fourth International Conference on Computer and Information Technology (CIT '04)*, September 2004, pp. 693–700.

[26] M. Sharmin, S. Ahmed, and S. I. Ahamed, “MARKS (Middleware Adaptability for Resource Discovery, Knowledge Usability, and Self Healing) in Pervasive Computing Environments,” *Third International Conference on Information Technology: New Generations,* NV, USA, April 2006, pp. 306-313.

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

[28] S. Ahmed, M. Sharmin, and S. I. Ahamed, “GETS (Generic, Efficient, Transparent, and Secured) Self-healing Service for Pervasive Computing Applications,” *submitted*.

[29] S. Ahmed, M. Sharmin, and S. I. Ahamed, “PerAd-Service: A Middleware Service for Pervasive Advertisement in M-Business,”*29th International Computer Software and Applications Conference,* Edinburgh, Scotland, July 2005, pp. 17-18.

[30] OMNeT++ Community Site, URL: http://www.omnetpp.org/

[31] F. Zhu, M. Mutka, and L. Ni, “Classification of Service Discovery in Pervasive Computing Environments,” MSU-CSE-02-24, MSU, 2002, pp. 1-17.

[32] 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, Accessed May 2006.

[33] F. Almenarez, A. Marin, D. Dyaz, and J. Sanchez, “Developing a Model for Trust Management in Pervasive Devices,” *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*, 2006, pp. 267-271.

[34] P. R. Zimmermann, “PGP Source Code and Internals,” MIT Press, 1995.

[35] 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, USA, April 2005, pp. 271-276.

[36] M. Sharmin, S. Ahmed, and S. I. Ahamed, “An Adaptive Lightweight Trust Reliant Secure Resource Discovery for Pervasive Computing Environments,” *Fourth Annual IEEE Int. Conference on Pervasive Computer and Communications (PerCom 2006),* Pisa, Italy, March 2006, pp. 258-263.

[37] M. Sharmin, S. Ahmed, and S. I. Ahamed, “SSRD+: A Privacy-aware Trust and Security Model for Resource Discovery in Pervasive Computing Environment,” *Pro. of the 30th Annual International Computer Software and Applications Conference (COMPSAC 2006)*, Chicago, September 17-21, 2006.

Authorized licensed use limited to: Marquette University. Downloaded on December 7, 2009 at 15:03 from IEEE Xplore. Restrictions apply.