SSRD+: A Privacy-aware Trust and Security Model for Resource Discovery in Pervasive Computing Environment

Moushumi Sharmin, Sheikh I. Ahamed, Shameem Ahmed, and Haifeng Li
Department of Mathematics, Statistics and Computer Science
Marquette University, Milwaukee, Wisconsin, USA
{msharmin, iq, sahmed02, hli}@mscs.mu.edu

Abstract

SSRD is a secure resource discovery model for devices running in a pervasive computing environment. SSRD is based on a lightweight trust model. SSRD+ is an extension of the existing SSRD model. In SSRD+, we enhance the trust model by adding dynamic trust relationship and also specifying behavioral characteristics that determine the level of trust among devices. We also add a risk model to address challenges posed by the pervasive and ad hoc nature of the network. These models work together to make the entire discovery process lightweight and secure. In this paper we present details of the trust and risk models. We illustrate the design and implementation of SSRD+ as a whole that optimally explores resources without degrading the performance of the devices while ensuring user security and privacy.

1. Introduction

The pervasive computing environment is comprised of numerous devices that include PDAs, cell phones, smart phones, laptops, etc. Nowadays, these devices are truly everywhere making Weiser's vision a reality [1]. These devices interact with other devices in an ad hoc manner. Resource discovery is an essential part of devices running in a pervasive computing environment [10]. The resource discovery process demands models that ensure privacy and security of the user [2, 3, 4]. The traditional security mechanism does not work in this environment, as the devices are computationally poor and the notion of physical security is not applicable [5]. The concept of human trust is now being used as a tool of ensuring security and protecting user privacy in pervasive computing.

From a security viewpoint, resource discovery models can be divided into three broad categories. First are the resource discovery models that do not address security issues [11-15]. Secondly, there are models that consider a full-fledged security mechanism with the help of some fixed infrastructure support (powerful servers, proxies, etc.) [17-19]. Others support security with the assistance of hardware [20], authentication [21], and trust [22, 16]. In SSRD [6], we presented a trust based secure resource discovery model. Our model was designed for a truly pervasive environment, where we assume that the mobile devices would be able to handle necessary computations and communications by themselves without any fixed infrastructure support. This simple model allowed resource discovery and sharing based on mutual trust. However, for unknown devices building trust relationship is complicated and sometimes impossible. To handle situations like this, we feel that with trust model, a risk model should be added. The necessity of risk assessment for resource discovery is presented in [9]. In this paper, we present a new model SSRD+, which is an extension of our existing SSRD model. Here, we have modified the trust model to make the trust relationship more accurate. We also propose a risk model that allows unknown devices to get services.

The outline of this paper is as follows: We present the design and architecture of our proposed model in Section 2. The evaluation of our proposed model is presented in Section 3 followed by concluding remarks and future research direction in Section 4.

2. Design and Architecture

In this section, we present different models that comprise the SSRD+. We have added this to the existing SSRD model, which is a part of SAFE-RD [7], the resource discovery unit of MARKS [8]. The SSRD+ unit handles security related issues and consists of *trust management and risk assessment* 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. A detail description of the architecture can be found in [6, 8]. All these units provide for user 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. In this paper, we describe the newly added features of our trust model and our risk assessment model. Description of the other features with architectural detail of MARKS, SAFE-RD, and SSRD can be found in [6-8].

Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC'06)
0-7695-2655-1/06 $20.00 © 2006 IEEE

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: Marquette University. Downloaded on December 7, 2009 at 15:32 from IEEE Xplore. Restrictions apply.

2.1 Trust Model

In SSRD [6], we elaborately described our trust model with its trust properties. Here we are only presenting the additional features that we have added to make this trust model more efficient. A human-like trust relationship is desirable in a pervasive computing environment, but it is impossible to impose robust security among devices in this environment. In human societies, trust relationships are built on mutually agreed upon behavior and recommendations. Unknown persons also get an opportunity to become familiar with the recommendation of others and can enhance their trust level by displaying good behavior. Misbehavior also plays a role in determining the level of trustworthiness of a person. We wanted to design a model that is similar to the human trust relationship model. However, the difficulty here is determining what behaviors are rewarded and what are not acceptable. It is difficult to judge. In a smart space, when the users are known and have access rights associated with resources, then we can calculate the degree of positive or negative behavior by observing whether or not they are trying to access non-permitted devices, or whether they are dominating the use of a resource and causing problem for others. Similar criteria can be used to increase or decrease the level of trust in smart spaces. However, in a truly pervasive environment, where the entities do not have any associated privilege level or are totally unknown, finding criteria that increases or decreases the level of trust of a device is extremely difficult to fine-tune. For simplicity we divided behaviors into three broad categories – good or rewarding behavior (this increases the average trust value), neutral behavior (which does not have any effect on the average trust value), and negative behavior (decreases the trust value). To evaluate these behaviors numerically, we have stored the average service time *(this includes request time + service offer time)* in the risk table. Whenever a device requests a service and is offered that service, then the total time required to complete the service request is recorded and compared with this average time. The average trust level value is updated using the following formula:

$$u_i = (RT - AST) / AST \ldots\ldots\ldots (1)$$

$$t(SP, D) = t(SP, D) + u_i \ldots\ldots\ldots (2)$$

$$t(D) = \frac{\sum_{i=1}^{n} u_i * w_i}{n} \pm c \ldots\ldots\ldots\ldots (3)$$

Here,

u_i = Modification value for service i

$t(SP, D)$=Trust value of device D for service SP

$t(D)$ = Average trust value of device D

ω_i = 0.1* security level of service i

n = Number of services relating to provider and requester

RT = Required time for a successful request completion

AST = Average service time

c = Random behavioral parameter

In equation 3, c is included to reflect behaviors such as sending too many requests in a small amount of time, repeatedly sending the same request that is already rejected, and so on.

Our trust model is privacy aware. It consults the user in case of 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. 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.

2.2 Risk Model

A risk model is essential in 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. In SSRD, when a service request arrived, we calculated the trust value of the requesting device (if the provider device has information about the requester or by collecting recommendation from other devices). Then based on the security level of the requested service, we accepted or denied the request. When the requester was unknown to all the neighboring devices (a very common scenario in pervasive computing), the device was assigned an initial trust value 0.5 which would allow it to receive lower security-intensive services and build a trust relationship with others. However, if that device required a higher security level service, it was denied. To address this issue, we added the risk assessment model in our trust model.

This 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 accept or reject event, it updates the risk value associated with that service. It collects information about number of accepts, number of rejects, average trust values of the devices who request this service, service time, etc. Based on these values, it assigns a risk factor with the service. As this

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: Marquette University. Downloaded on December 7, 2009 at 15:32 from IEEE Xplore. Restrictions apply.

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 1 shows some sample data stored in a device-

Table 1. Risk value table

Id	Number of request	Number of accept	Avg. trust value	Avg. service time (ms)
5	3	1	0.75	21
9	7	6	0.6	15
13	17	13	0.83	40
…	…	…	…	…

For each service there number of requests, number of accepts, average trust value of devices for which the request is accepted, and average service time to offer this service is stored. All these data are used to calculate the risk factor for sharing a service. If the percentage of acceptance is greater than 50% and the average trust value is around 0.6 for a service (i.e. we offer this service to devices with moderate trust level), for unknown devices the service is shared. 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.

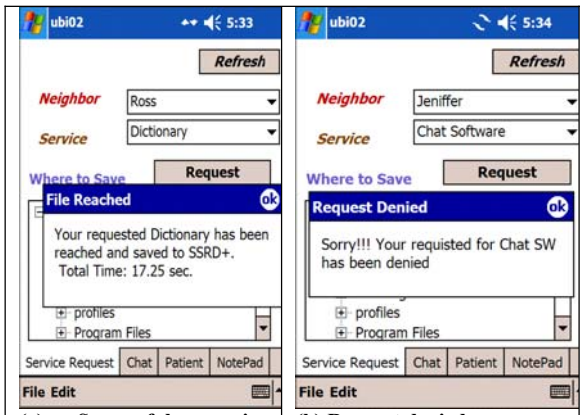
3. Evaluation

To evaluate the effectiveness of our SSRD+ model we have used the following techniques:

1. Prototype implementation
2. Performance measurement

3.1 Prototype Implementation

To implement the additional features of SSRD+, we developed an application prototype using a test bed consisting of a set of Dell Axim X50v 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 the laptop, desktop, and smart phone. As the underlying wireless protocol, we used the mobile ad hoc mode of IEEE 802.11b.



(a) Successful service request (b) Request denied

Figure 2. Application that uses SSRD+

3.2 Performance Measurement

We tested the performance of our SSRD+ model and compared it with the SSRD model. Table 2 lists the service time comparison of SSRD and SSRD+. Services with lower security levels do not experience any response time change since the services' risk model is never used. From our experiment we found the services that require higher response time also are not affected since the risk evaluation time is very negligible compared to overall service response time. From the results it is proven that the added features do not result in much computation overhead. Some screenshots of the prototype are shown in Figure 2. Since all the required changes in the model work on backend, we did not change application front-end design that we used for SSRD. However, to check the required service response time we only added the time value in a message box.

Table 2. Comparison of service time

	Time (Sec)		
Service Name	**Normal**	**SSRD**	**SSRD+**
DateTime	0.1	0.105	0.105
WAV (148KB)	0.7	0.72	0.75
Chat SW (262KB)	0.9	0.925	0.975
Unzip SW (323 KB)	1.0	1.03	1.07
Address Service (810KB)	1.8	1.91	1.91
Dictionary (5.94MB)	17.2	17.25	17.25
Music SW (7.96MB)	23.6	23.66	23.66
Acrobat Reader (13.5 MB)	40	40.05	40.05

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: Marquette University. Downloaded on December 7, 2009 at 15:32 from IEEE Xplore. Restrictions apply.

4. Discussions and Future Work

In this paper, we have proposed SSRD+, which is an extension of the existing SSRD model. We identified the need for a risk assessment model with differing levels of security for different services. Our aim was to provide a privacy-aware trust-based lightweight security model that works in truly pervasive environments. We designed a simple yet efficient risk based trust model that handles security related issues without causing too much battery power consumption. We also evaluated the performance of our model using prototype implementation. Currently, our risk model uses historical data to calculate the risk factor. We are examining different distributions to find out the threshold value for risk parameters that minimizes the risk of service sharing. In our future work, we will make our model more robust by including appropriate risk parameters.

References

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

[2] F. Stajano, "Security for Ubiquitous Computing," pp. 110-111, *Wiley*, ISBN 0-470-84493-0.

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

[4] K. Matsumiya, S. Tamaru, G. Suzuki, J. Nakazawa, K. Takashio, and H. Tokuda, "Improving security for ubiquitous campus applications," *SAINT 2004*, Jan 2004, pp. 417–422.

[5] L. Kagal, T. Finin, and A. Joshi, "Moving from Security to Distributed Trust in Ubiquitous Computing Environments," *IEEE Computer*, Dec 2001.

[6] M. Sharmin, S. Ahmed, and S. I. Ahamed, "An Adaptive Lightweight Trust Reliant Secure Resource Discovery for Pervasive Computing Environments," *PerCom2006*, Pisa, Italy, Mar 2006, pp. 258-263.

[7] M. Sharmin, S. Ahmed, and S. I. Ahamed, "SAFE-RD (Secure, Adaptive, Fault Tolerant, and Efficient Resource Discovery) in Pervasive Computing Environments," *ITCC 2005,* USA, Apr 2005, pp. 271-276.

[8] M. Sharmin, S. Ahmed, and S. I. Ahamed, "MARKS (Middleware Adaptability for Resource Discovery, Knowledge Usability, and Self Healing) in Pervasive Computing Environments," *ITNG06,* USA, Apr 2006, pp. 306-313.

[9] 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.

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

[11] M. Nidd, "Service Discovery in DEAPspace," *IEEE Personal Communications*, Aug 2001, pp. 39-45.

[12] B. A. Miller, T. Nixon, C. Tai, and M. D. Wood, "Home Networking with Universal Plug and Play," *IEEE Communications Magazine*, Dec, 2001, pp. 104-109.

[13] 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, SC, 1999.

[14] 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, 2002.

[15] Microsoft Corporation, "Universal Plug and Play Device Architecture," Version 1.0, Microsoft Co., 2000.

[16] 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)*, Sep 2004, pp. 693–700.

[17] 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.

[18] F. Zhu, M. Mutka, and L. Ni, "Splendor: A secure, private, and location-aware service discovery protocol supporting mobile services," *Pervasive Computing and Communications,* Mar 2003, pp. 235–242.

[19] F. Zhu, M. Mutka, and L. Ni, "PrudentExposure: A Private and User-centric Service Discovery Protocol," *Second IEEE International Conference on Pervasive Computing and Communications (PerCom 2004),* Mar 2004, pp 329-340.

[20] 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, Mar 2005, pp. 199-203

[21] F. Zhu, M. Mutka, and L. Ni, "Expose or not? A progressive exposure approach for service discovery in pervasive computing environments," *Percom 2005*, Mar 2005, pp. 225–234.

[22] 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, Sep 2003.

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: Marquette University. Downloaded on December 7, 2009 at 15:32 from IEEE Xplore. Restrictions apply.