Security for Ubiquitous Computing: Problems and Proposed Solution

Le Xuan Hung, Tran Van Phuong, Pho Duc Giang, Yonil Zhung, Sungyoung Lee and Young-Koo Lee
Department of Computer Engineering, Kyung Hee University, Korea
{lxhung,tvphuong, pdgiang, zhungs,sylee}@oslab.khu.ac.kr, yklee@khu.ac.kr

Abstract

Traditional authentication and access control are no longer suitable for ubiquitous computing paradigm. They are only effective if the system knows in advance which users are going to access and what their access rights are. Therefore, it calls for a novel security model. In this paper, we outline major security problems in ubiquitous computing and propose a new architecture, TBSI (Trust-based Security Infrastructure). In TBSI, trust and risk management plays a key role to support authentication and authorization to unknown users. Meanwhile, intrusion detection and home firewall are also integrated to make TBSI more robust. This paper is an extension of our previous work to give more detailed description and enhancement of the architecture. TBSI is on-going research project to support our context-aware middleware CAMUS. [1]

1. Introduction

"*Ubiquitous computing*" technology has started since M. Weiser introduced this new term in his paper [1]. This new technology envisions a world where embedded processors, computers, sensors, and digital communications are inexpensive commodities that are available everywhere. The technology is '*disappeared*' into background and provides users services '*anywhere at anytime*'.

However, ubiquitous computing raises many security, privacy and trustworthy problems. It requires a security architecture based on trust rather than just user authentication and access control that traditional, stand-alone computers and small networks rely on [2]. This is because authenticating the identity certificate of a previous unknown user does not provide any access control information. Simple authentication and access control are only effective if the system knows in advance which users are going to access the system and what their access rights are.

In this paper, we address major issues in ubiquitous security design and propose our TBSI architecture. TBSI is based on trust level of an unknown entity to authenticate and to make decision whether to allow or deny her access right. On the other hand, intrusion detection and home firewall are integrated to make TBSI more robust. This paper is an extension of our previous work [3] in which we provide more detailed description as well as enhancement of TBSI. The main contribution of this work is to investigate the security threats in ubiquitous computing environments and to propose a well-designed solution based on notion of trust. It has solved one of the major challenges in ubiquitous computing that unknown entities (including users, devices, applications, etc) can not be authenticated and authorized by conventional security systems. Another contribution of this work is that it provides flexible and scalable security mechanism and adapts well with ubiquitous computing environments.

The rest of this paper is organized as following. In next section, we briefly talk about major security problems in ubiquitous computing. Section 3 shortly mentions about related work. We describe TBSI architecture and its module design in Section 4. Section 5 concludes the paper and outlines our future work.

2. Security Problems in Ubiquitous Computing

Traditional computing security is based on authentication and access control to authenticate and authorize users who have already registered to the system. This means that a user has to possess an authenticated certificate which is corresponding to some certain access right. However, these approaches are no longer suitable for increased flexibility and scalability of ubiquitous computing. This is because there are a huge number of entities and those entities are not always predetermined. How can a system decide whether a user who does not belong to an office but has a right access to it?

[1] This work is financially supported by the Ministry of Education and Human Resources Development (MODE), the Ministry of Commerce, Industry and Energy (MOCIE), and the Ministry of Labor (MOLAB) through the fostering project of the Lab of Excellency. Dr. Sungyoung Lee is the corresponding author.

Another important problem is related to network infrastructure of ubiquitous computing. In order to make the system satisfy *invisibility* and *ubiquity*, i.e. the technology disappears into background while supporting users to access or to use services anywhere at anytime, the ubiquitous network nodes, such as mobile devices or sensing devices, should be wirelessly and seamlessly interact and collaborate with each other. Those devices and services are prone to be compromised or attackers can easily insert some forge nodes to attack the system. A firewall solution and an intrusion detection system are necessary to protect ubiquitous network infrastructure in such cases.

3. Related Work

Though ubiquitous computing was started almost two decades ago, its security has just been addressed in recent years and no work has been completed. In [2], L. Kagal *et al* early introduced a trust-based security architecture for pervasive computing. In this paper, the authors agued that pervasive computing systems require security architecture based on trust rather than just user authentication and access control. However, they did not provide any solution as well as concrete model. Al-Muhtadi *et al* [4] introduced a context-aware security scheme for smart spaces (Cerberus). This is a dynamic, context-aware security service supporting GAIA middleware [5]. Cerberus features a federated authentication system that is based on distributed, pluggable, "*CORBArized*" authentication module. However, Cerberus would not work for old applications not using GAIA. This is also problem of other security services that follow middleware-based approach. Another approach, SECURE [6] was proposed by J. M. Seigneur *et al*. The heart of the SECURE project is the development of a computational model of trust that will provide the formal basis for reasoning about trust and for the deployment of verifiable security policies.

4. TBSI Design

TBSI is a component-based infrastructure which includes five major modules: authentication (PRM-Pluggable Recognition Module), access control (FSAC-Flexible and Scalable Access Control), trust management (TMBV-Trust Management Based on Vector of trust values), home firewall (HFW), and intrusion detection system (IDS). The general architecture is depicted in Fig 1.

We classify ubiquitous network nodes into two categories: authentication devices which are equipped by users, and sensing devices which gather contextual information from environment. At the network layer, TMBV manages and delegates trust between nodes to support their interactions and collaborations. IDS can detect misbehavior or abnormal actions in sensor nodes. HFW is deployed between those network devices and context-aware middleware to eliminate at maximum attacks from network infrastructure. Users and sensing devices must be authenticated by PRM and authorized by FSAC.

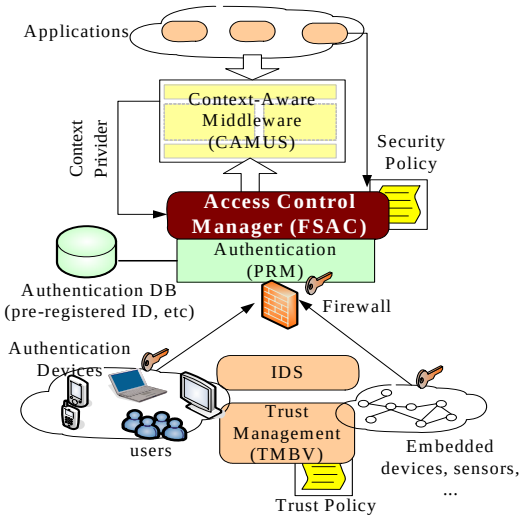

Figure 1. TBSI Architecture

Fig 2 depicts the protocol of FSAC including three major operations: *deduction*, *abduction*, and *trust evaluation*. When a user sends a service request (REQ) to the ACM (Access Control Manager) along with his credentials (C_P), the ACM firstly perform *deduction* operation (step 1, written in the circle). It evaluates the service request by using policy rules from Policy Manager (PM) and context from Context Provider (CP), and it makes a decision whether this request is permitted or not. If the service request is not allowed due to limited privilege of his role and current credentials, the system will pass this request to *abduction* operation. By checking the request and system policies, the abduction operation is performed to find the minimum additional credentials (C_m) which the user must provide additionally in order to gain access. This additional credential requirement is sent to the user (step 2). If the user can provide such credentials, then there is no problem for him to access the resource (step 3).

So far we have presented the cases in which the user is determined by the system and he can provide sufficient credentials to the system. However, such cases do not cover all the

circumstances in ubiquitous computing environments. Usually, the user is unknown by the system and he is not able to provide such required credentials that the system requests. FSAC deals with this problem by supporting Trust Comparison (TC) operation. Trust Calculator computes trust value on this unknown entity (T_V) based on recommendations of other entities, history of its interaction with the system, and other. To do this, services, resources, and entities must maintain some latest information of interaction with the unknown entity. Also, though this entity is unknown to the system, it must be able to be authenticated by entities who are going to recommend it. It then passes this value to Trust Comparison (step 5). If the trust value is greater the predefined trust value of given service/resource, he will be permitted to access to that service/resource, otherwise denied. After making decision, ACM sends a feedback to Trust Calculator to update interaction information (step 6).

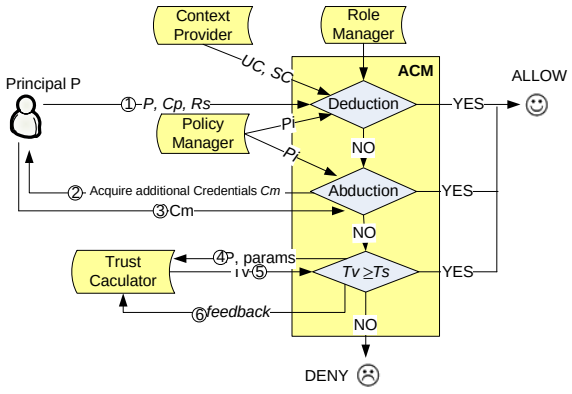

Figure 2. FSAC Protocol

In FSAC, *trust* of principal Q_i on principal Q_j is calculated based on a vector of trust values [7] including *Peer Recommendation (PR), Confidence level (CF), History of Past Interaction (PI), and Time-based Evaluation (TE).*

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

where w_1 and w_2 is weighted value that can be adjusted to meet different security requirements of various systems[2].

[2] Refer [7] for more details about our Trust Model,

In the Intrusion Detection System design we deploy a *Dynamic Agent* (DA), as depicted in Fig 3, in each sensor node to monitor abnormal behaviors of itself and neighboring nodes, and report to the IDS server. Due to resources limitation of sensor nodes, this DA is lightweight and dynamic with simple functions supporting IDS server.

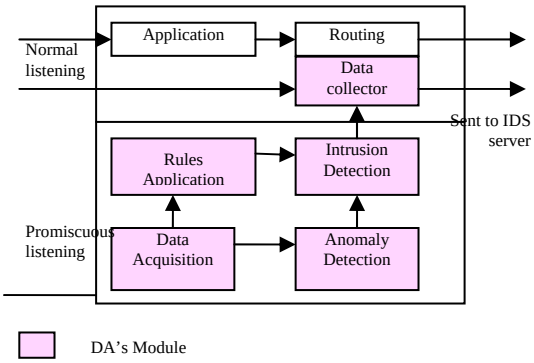

Figure 3. Sensor's DA Architecture

In *DA*, *Data Acquisition* is in charge of collecting message in *promiscuous* mode and important information is filtered before being stored, for subsequent analysis. *Rule Application* is the processing module, at which the rules are applied to the stored data. If the message analysis fails the tests being applied, a failure is raised. To save energy and memory of sensor node, the set of rule in this module is dynamic. This means each sensor has a function to support a certain application. When the function of sensor is changed, then the sensor is notified by IDS server to remove all previous rules and download a new set of rule on IDS server corresponding to new application's function. *Anomaly Detection* detects anomaly behaviors based on statistical data of packets. *Intrusion Detection* compares the number of current failures with the number of occasion failures in the network. If the number of current failures is higher than a trigger of intrusion detection will perform to notify *Data Collector*. This module collects and sends necessary data to the IDS server for further processing. In the sake of saving resource, this module is active only when the *DA* detects something abnormal in the network. IDS server is responsible for collecting all data from necessary sensor nodes, aggregating, and making the final intrusion decision. In our model, IDS server can use both signature based and anomaly detection technique.

Home Firewall is introduced to protect central server as well as network infrastructure in smart space. The major functions performed by the

HFW are to screen incoming traffic and block suspicious code, to screen outgoing messages that infect other resources, and to prevent the unauthorized use of logical ports by hiding them from malicious code or human penetration attempts. In our approach, HFW contains MAC Address Refining (MAR) module. This module is responsible for real-time selecting trusted MAC addresses of available confident base stations in the space for preventing base station spoofing attack. The selected addresses are maintained in an admission list. The MAR module periodically sends a RARP (Reverse Address Resolution Protocol) packet to each address in the list. The function of RARP is mapping a MAC address into an IP address. Following this, Reverse ARP should reply one IP address for one network device. If multiple IP addresses return, it means that the MAC address is being exploited by more than one device. The firewall manages all the transactions between the user's mobile devices and the central server. If the WAP and/or user's mobile device are compromised, attackers still have no way to change the behavior of our server since they don't know the username/password to change the firewall policies. Our policy, i.e., was set to turn the camera system on from 11P.M to 6A.M. Therefore, malicious control packets that want to improperly turn the system off at that time will be dropped by the firewall. The Home Firewall also helps preventing other server's programs from being compromised by stopping common hacker's reconnaissance port scanning techniques. In order to defend our server from these kinds of potential threats, such as ICMP scanning, TCP scanning, UDP scanning, we deploy an anti-scanning security policy. Our firewall will prohibit the ICMP replying packets for preventing ICMP scanning technique and deny the ICMP Port Unreachable packets transmitted back to an attacker for protecting UDP scanning probe. For detecting the TCP scanning signature, we might say that if there are more than 5 SYN packet attempts to non-listening ports in one minute, an alarm SMS message should be automatically triggered to the user's cell phone.

6. Conclusion and Future Work

In this paper, we have argued that the security architecture for this technology must be based on trust, rather than just user authentication and access control that traditional stand-alone computers and networks have relied on. We also presented a component-based security infrastructure TBSI to tackle most of security problems in ubiquitous computing systems. Each module is designed to be lightweight enough to fit limited resources of sensing and mobile devices.

Though TBSI is a promising model for next generation of ubiquitous security, there are a lot of work needed to be completed. At the current stage, we have finished TBSI model and module design. We are working on security policy specification including access control policy, firewall policy, and IDS policy by adapting Ontology Web Language (OWL) and Prolog. TBSI is currently being deployed for smart space. After accomplishing in this environment, we will extend to other environments such as parking spaces, airports, and hospitals.

References

[1] M.Weiser, "Hot Topics: Ubiquitous Computing" IEEE Computer, 1993

[2] Lalana Kagal, Tim Finin, and Anupam Joshi. Trust-Based Security in Pervasive Computing Environments. In: IEEE Computer, pages 154–157, Dec 2001

[3] Le Xuan Hung, Pho Duc Giang, Yonil Zhung, Tran Van Phuong, Sungyoung Lee and Young-Koo Lee. A Trust-based Security Architecture for Ubiquitous Computing Systems. IEEE Intelligent and Security Informatics (ISI-2006), May 23-24 San Diego. LNCS 3975 / 2006 pp. 753 - 754 .

[4] Jalal Al-Muhtadi, Anand Ranganathan, Roy Campbell, and M. Dennis Mickunas. Cerberus: A Context-Aware Security Scheme for Smart Spaces. In IEEE International Conference on Pervasive Computing and Communications (PerCom 2003), Dallas-Fort Worth, Texas, March 23-26, 2003

[5] Manuel Román, Christopher K. Hess, Renato Cerqueira, Anand Ranganathan, Roy H. Campbell, and Klara Nahrstedt. Gaia: A Middleware Infrastructure to Enable Active Spaces. In IEEE Pervasive Computing, pp. 74-83, Oct-Dec 2002.

[6] Jean-Marc Seigneur, Stephen Farrell, Christian Damsgaard Jensen. Secure ubiquitous computing based on entity recognition. Ubicomp2002 Security Workshop 2002.

[7] Hassan Jameel, Le Xuan Hung, Umar Kalim, Ali Sajjad, Sungyoung Lee and Young-Koo Lee. A Trust Model for Ubiquitous Systems based on Vectors of Trust Values. 3rd International IEEE Security in Storage Workshop San Francisco, California USA, Dec 13, 2005