Design of a Security Management Middleware in Ubiquitous Computing Environments

Zhefan Jiang[1], Kanghee Lee[1], Sangok Kim[1], Hyunchul Bae[1], Sangwook Kim[1], Soongju Kang[2]
Department of Computer Science[1],
School of Electrical Engineering and Computer Science[2],
Kyungpook National University
1370 Sankyuk-Dong, Buk-Gu, Daegu, 702-701, Korea
{zfjiang, sokim, khlee, hcbae, swkim}@cs.knu.ac.kr[1], sjkang@ee.knu.ac.kr[2]

Abstract

In this paper, we propose a security management middleware in ubiquitous computing environments. This security management should be application-oriented security and provides user-centric security. This middleware provides the services for user or administrator to define security policies which can reflect dynamic context and trust management. We define a new security specification policy language for policy management. It also provides management services which can monitor and control the managed objects and sensors for the domain security management.

1. Introduction

In ubiquitous computing environments, users expect to access resources and services anytime and anywhere, so there is a need for more automated and secure management in ubiquitous computing formed by users accessing these resources and services[1].

The middleware architecture should provide selective domain-specific management and be adaptive for different domain management. When users take their mobile devices like PDA to access resources or services across domains, they should be authenticated, and be checked the permissions and monitored their activities.

Context-awareness is necessary for providing more flexible security to capture user's location or other context information and used to provide augmented services. Trust was defined as "a means to reason about and accept risk in situations of partial information and assign privileges accordingly." The trend in trust management systems is to view trust implicitly through the delegation of privileges to trusted entities via the use of certificates, which is based on both identity and context. Ubiquitous computing needs application-oriented security, therefore, a new adaptive security management model is required.

In this paper, we propose a security management middleware in ubiquitous computing environments. This middleware provides policy management, object management, context management, status monitoring and authentication management service.

The remainder of this paper is structured as follows. Section 2 describes existed related work. Section 3 shows our middleware architecture and each component. An experiment scenario using a printing application is presented in Section 4. Finally, we draw some conclusions and outlines directions for future research.

2. Related Work

Several research efforts have addressed the general issue of middleware solutions to support security and management in ubiquitous computing environments.

In CASA, they proposed secure service architecture for context-aware environments. They focus on the design of security services that incorporate the use of security relevant "context" to provide flexible access control and policy enforcement[2]. The Gaia project is a distributed middleware infrastructure that provides support for ubiquitous computing. The Cerberus core service of Gaia integrates identification, authentication, context awareness, and reasoning[3]. In Ubiquitous Context-based Security Middleware(UbiCOSM), a context-centric security middleware dynamically determines the contexts of mobile proxies, and rules the access to them[4].

These approaches were focused on the context-aware access control and authentication for ubiquitous

Proceedings of the Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'05)
0-7695-2405-2/05 $20.00 © 2005 IEEE

applications and management system. However, however, only authentication and access control is not insufficiency. And these middleware architectures were not concerned about trust management and lack to support real-time mobile application scenarios.

3. Security Management Middleware

The security management middleware architecture we proposed is shown in Figure 1.

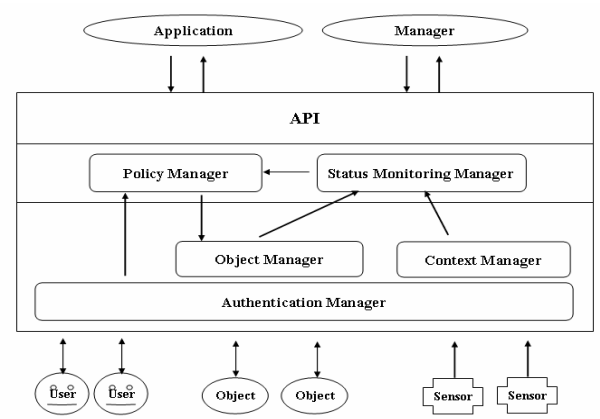

Figure 1. Middleware Architecture

3.1. Policy Manager

Policy-based security management is often used in systems where flexibility is required. Usually, security policy is defined to be the set of more or less informal rules that specify how the access control, confidentiality and availability of the stored information and available resources are to be protected within computing system.

The policies involve manual operations and management conventions that are expected to be followed by humans. We define three kinds of policies used in our policy management service shown in figure 2[5]. They are authorization policy, delegation policy, obligation policy.

```
Authorization Policy:
   <Subject, Object, Operations, Context, Perm>
Delegation Policy:
   <Sender, Receiver, Object, Operations, Conditions>
Obligation Policy:
   <Subject, Object, Events, Conditions, Actions>
```

Figure 2. Policy Specification

Authorization policy: it defines access control policy. And it reflects the context as a parameter for access control. Subject can be either a specific user or a role. Object can be target objects or a group name of the objects. A context is a collection of related variables, such as time or location. Sign has two values: grant and deny.

Delegation policy: A delegation allows a user to transfer its right to another user. It is used for trust management. Sender can delegate the privileges or rights to receiver when the conditions are true.

Obligation policy: it is actions that an object must perform and are usually event-driven. When a certain set of conditions are true. When the status monitoring manager triggers event from sensors or objects activities, the policy manager checks the conditions and enforces correspondent actions.

Policy manager provides the interface for domain administrator to define policies. These policies are encoded into XML and stored in policy repository.

3.2. Object Manager

Object manager provides to monitor and control target objects. We treat each device(ie., PDA, laptop or security related application) as an object. Object manager has two core functions. First is analyzing and integrate status information of objects. Second is managing the objects securely. It can catch the status of objects and configure parameters of objects for security. It has a unique ObjectID that created by object manager for clearly mark and delivery request message to status monitoring manager. And it also notifies available access to resources and receives resource status.

3.3. Context Manager

The main function of context manager is that collecting contexts from sensors and aggregating context data and send to status monitoring Manager. Communication manager provides different communication mode with various sensors. Distribute Processing manager is processing the distribute sensors efficiently. Information analyzer and filter manager analyzes continuous data from sensors and filters it. Context manager deliveries the control message from policy manager to the sensors which can control.

3.4. Status Monitoring Manager

Status monitoring manager collect the status information from object manager and context manager. Status information analyzer analyzes collecting data and send to DB manager to log. Filter manager filters redundancy information to analyze efficiently. Event

manager triggers event or context and send to policy manager to find the matching policy.

3.5. Authentication Manager

There are many methods to support identification like id-password, bio-information or security card. But in this middleware, the identify method is used as X.509 credential. Thus authentication manager provides user and object identification service, credential repository management and delegation service. When a user accesses the resource, the system inspects the credential information with credential that is stored in the repository. To support credential mechanism, authentication manager provides several credential services like generating credential, destroying credential, searching and updating credential in repository. And it also provides delegation service for support mobility. Delegation service uses composite delegation model. It uses delegated credential that combines it's own with other's.

4. Experiment

In this section, we describe the operation process of the proposed middleware architecture using a printing service scenario. In one company, Mike is an employee, he want to print his file in his PDA using the printer which in conference room. When he sent the printing request message which consisted User id, Device id, target printer name, and the print operations. If the authentication was successful, the object manager received messages and obtained access to printer. If the context manager had the related context data about Mike or his PDA, the object manager made message format of request adding context information and deliver to the policy manager.

We assumed that there was one authorization policy as following in policy repository:

<Subject: "Mike"; Object: "Printer"; Operations: "print"; Context: "location(ConferenceRoom)"; Perm: "grant">.

The policy manager analyzed this request message and then checked each parameter and authorized Mike to access printer and printed his files.

We've implemented our middleware using CORBA technology. CORBA offers several services that are instrumental in implementing some of the needed functionality. In object manager, we used static object invocation for resource objects and used dynamic invocation interface to manage the mobile devices. We also used CORBA object naming service and event services to manage and control the objects. Each component of this middleware was defined using Interface Definition Language(IDL), so users can select these Interfaces to develop the application which they need.

5. Conclusion and Future Work

In this paper, we proposed a security management middleware for ubiquitous computing environment. This middleware provides the services for user or administrators to define security policies which can reflect dynamic context and trust management. We define a new security specification policy language for policy management. It also provides management services which can monitor and control the managed objects and sensors for the domain security management. The future works are building a school-level hierarchy domain for evaluating our middleware.

6. Acknowledgements

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 Assessment).

6. References

[1] L. Kagal, T.Fini and A. Josh, "Moving from Security to Distributed Trust in Ubiquitous Computing Environments," LNCS 2867, 2003.

[2] M. J. Covingtony, P. Fogla, Z. Zhan, "A Context-Aware Security Architecture for Emerging Applications Proceeding of the Annual Computer Security Applications Conference, Dec. 2002.

[3] M. Román et al, "Gaia: A Middleware Infrastructure to Enable Active Spaces," IEEE Pervasive Computing, pp. 74-83, Oct-Dec. 2002.

[4] A. Corradi, R. Montanari, D. Tibaldi, A. Toninelli, "A context-centric security middleware for service provisioning in pervasive computing," Applications and the Internet 2005, Proceedings. pp. 421-429, Jan. 2005.

[5] N. Damianou, "Ponder: A Language for Specifying Security and Management Policies for Distributed Systems," Workshop on Policies for Distributed Systems and Networks, Bristol UK, pp. 18-39, Jan. 2001