XACML 기반 홈 네트워크 접근제어 시스템의 설계 및 구현

이 준 호[†] · 임 경 식[††] · 원 유 재[†††]

요 약

홈 네트워크가 활성화되기 위해서는 보안 서비스의 제공이 필수적이며 특히 사용자에 대한 접근제어는 안전하고 차별화된 홈 네트워크 서비스의 제공을 가능하게 한다. 그러나 기존의 홈 네트워크 보안 기술은 접근제어를 거의 고려하지 않거나 특정 미들웨어에 종속적인 구조를 가진다. 따라서 본 논문에서는 상호 호환성 및 확장성이 뛰어난 차세대 접근제어 표준인 eXtensible Access Control Markup Language(XACML)를 이용하여 홈 네트워크에서 통합적인 접근제어를 제공하기 위한 방안을 제시하고 이를 바탕으로 XACML 접근제어 시스템을 설계하고 구현한다. 또한 구현된 XACML 접근제어 시스템을 OSGi기반 UPnP 프락시 시스템에 적용하여 다양한 정책에 대한 실험을 수행함으로써 기존 홈 네트워크 시스템과의 호환성을 검증하였다.

키워드 : XACML, 접근제어, 홈 네트워크

Design and Implementation of Access Control System Based on XACML in Home Networks

Junho Lee[†] · Kyungshik Lim[††] · Yoojae Won[†††]

ABSTRACT

For activating home network, the security service is positively necessary and especially the access control supports secure home network services and differentiated services. But, the existing security technology for home network seldom consider access control or has a architecture to be dependent on specific middleware. Therefore, in this paper we propose a scheme to support integrated access control in home network to use XACML, access control standard of next generation, to have compatability and extensibility and we design and implement XACML access control system based on this. we also had an access control experiment about various policy to connect developed XACML access control system with the UPnP proxy based on OSGi in order to verify compatability with existing home network system.

Key Words : XACML, Access Control, Home Network

1. 서 론

정보가전 기기가 하나의 네트워크를 구성하여 서로 간의 정보를 공유하고 유·무선 인터넷과 연결되어 집 내부 또는 외부에서의 다양한 홈 서비스를 가능하게 하는 홈 네트워크 기술은 유비쿼터스 환경을 위한 중요한 요소이다. 이러한 홈 네트워크의 활성화를 위해서는 홈 네트워크 환경에 적합한 보안 기술이 반드시 필요하다[1]. 이는 인터넷과의 연동이 필수적인 홈 네트워크에서 기존 인터넷에 존재했던 모든 보안상의 위험 및 문제점들이 발생할 가능성이 충분히 있으며, 접근 주체가 가정의 구성원이라는 점과 접근 대상이 되는 홈 디바이스의 종류 및 특성이 다양하기 때문이다[2]. 일반적인 보안 기술에는 기밀성, 무결성, 인증, 부인방지, 접근제어 등이 있지만 안전한 홈 네트워크 서비스를 위해서는 기본적으로 인증 및 접근제어 서비스가 제공되어야 한다. 특히, 접근제어는 인증된 사용자에 대하여 홈 서비스에 대한 접근 권한을 검증하기 때문에 차별화된 서비스를 제공하고 댁내 정보를 보호해준다.

대표적인 홈 네트워크 미들웨어 표준에는 UPnP[3], JINI[4], HAVi[5] 등이 있으며 서비스를 위한 플랫폼으로서 개방형 구조를 가지는 OSGi[6]가 있다. UPnP는 IP기반의 P2P방식을 지원하는 제어 프로토콜이며 디바이스 자체의 Access Control List(ACL)를 이용한 접근제어 방식을 UPnP 보안 표준에서 정의하고 있다[7]. JINI는 홈 네트워크 서비스 환경에 적합한 하부 구조를 정의하고 있으며 기밀성, 무결성 및 서비스에 대한 인증 및 접근제어 등의 보안 기능

† 정 회 원 : 한국전자통신연구원 정보보호연구단 홈네트워크보안연구팀
†† 정 회 원 : 경북대학교 전자전기컴퓨터학부 부교수
††† 정 회 원 : 한국정보보호진흥원 IT 기반보호단 응용기술팀장
논문접수 : 2005년 9월 13일, 심사완료 : 2006년 7월 10일

을 제공한다. HAVi는 IEEE1394 기술을 통한 AV기기 간 실시간 데이터 전송 및 상호 호환성을 위한 미들웨어이며 서비스에 대한 인증 및 접근제어 기능을 제공한다. OSGi는 서비스를 위한 미들웨어로서 번들의 설치 및 서비스 참조에 대한 보안 기술은 제공되지만 서비스를 이용하는 사용자에 대한 보안 기술은 제공하지 않는다.[8] 각 미들웨어는 기본적인 보안 기능들을 포함하고 있고 관련 보안 기술에 대한 표준화도 이루어지고 있지만, 사용자에 대한 인증 및 접근제어를 위한 기능은 아직 미흡하고 특정 미들웨어나 플랫폼에 종속되는 단점이 있다. 따라서 본 논문에서는 이러한 단점을 극복하고 특정 미들웨어에 종속되지 않는 접근제어 서비스를 제공하기 위하여 홈 게이트웨이에서 통합적인 관리가 가능한 접근제어 시스템을 설계하고 구현한다. 이는 XML 기반의 접근제어 표준인 XACML[9] 기술을 사용함으로써 정책의 가독성, 확장성, 호환성이 높고 특정 미들웨어에 종속되지 않으며 다양하고 섬세한 접근제어 기능을 제공한다. 또한 RBAC 접근제어 방식을 지원하기 때문에 복잡하지 않고 체계적인 접근 제어가 가능하며 차세대 통합 인증을 위한 표준인 Security Assertion Markup Language (SAML)[10]와의 연동이 가능하다는 장점이 있다.

XACML 접근제어 시스템의 개발 환경은 OSGi 프레임워크, UPnP 프락시 시스템, UPnP 프락시 관리 번들, XACML 접근제어 번들로 구성된다. OSGi 프레임워크는 홈 게이트웨이에 존재하여 서비스의 관리 및 연동 기능을 제공하며 UPnP 프락시 시스템은 댁내 또는 외부에서 원격으로 홈 디바이스를 제어하기 위한 시스템이다. UPnP 프락시 시스템을 관리하고 제어하는 역할을 하는 UPnP 프락시 관리 번들과 접근제어 기능을 제공하는 XACML 접근제어 번들은 OSGi 프레임워크에 탑재되어 동작된다. OSGi 프레임워크에 존재하는 모든 번들은 서로 간에 서비스를 이용할 수 있기 때문에 UPnP 프락시 관리 번들은 XACML 접근제어 번들을 이용하여 권한 검증을 수행한 후, UPnP 프락시 시스템으로 권한 검증의 결과를 전달한다. 이러한 접근제어 방식은 홈 게이트웨이를 중심으로 이루어지는 중앙 집중형 방식이기 때문에 보안 관리의 측면에서 안정적이고 편리하며, 다양한 디바이스 및 미들웨어에 대한 통합적인 접근제어 및 보안 기술의 적용이 가능하다. 이는 통합 미들웨어 환경 및 대규모 홈 네트워크 환경에 적합하며, 다양한 서비스 및 미들웨어에 대한 접근제어 관리가 용이하다는 장점이 있다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 접근제어 기법 및 XACML에 대하여 살펴보고 3장에서는 XACML 접근제어 시스템의 구조 및 접근제어 동작 과정에 대하여 설명한다. 4장에서는 구현 내용에 대하여 상세히 기술하고 5장에서는 실험 내용 및 결과를 설명하며 마지막으로 6장에서 본 논문의 결론을 맺는다.

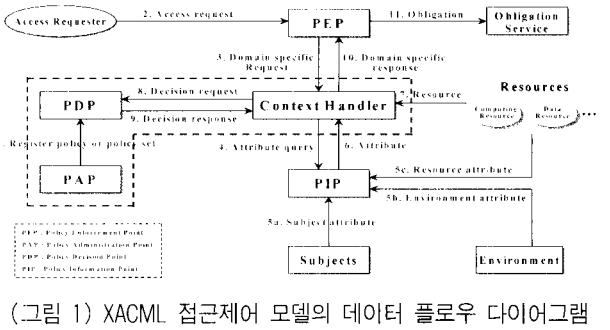
2. 관련 연구

본 장에서는 e-Business에서의 차세대 접근제어 표준인 XACML에 대하여 살펴보고 주요 기능 및 특징에 대하여 설명한다. 그리고 기존의 접근제어 관련 기술로서 Secure OS에서의 접근제어 방식 및 내부 구조를 설명하고 자바2 플랫폼에서 사용하는 접근제어 메커니즘의 특성에 대하여 설명한다.

2.1 XACML

XACML은 정책을 이용하여 보안이 요구되는 자원에 대해 접근제어 서비스를 제공할 수 있는 XML 기반의 언어이다. 이는 SAML의 PDP(Policy Decision Point)로서의 역할을 수행할 수 있으며 DAC, MAC, RBAC등의 다양한 접근제어 기법을 지원할 수 있다. XACML의 장점은 접근에 대한 허가 또는 거부와 같은 단순한 결과를 반환하는 것이 아니라 자원에 접근하기 위한 여러 가지 조건을 명시할 수 있으며 이에 따른 다양한 결과를 반환하기 때문에 미세한 접근제어 서비스가 가능하며 접근제어를 위한 정보 교환의 표준을 제공함으로서 다양한 시스템간의 상호 호환성을 제공한다.

XACML 접근제어 모델은 (그림 1)과 같이 정책의 설정 및 관리 등을 담당하는 정책 관리 포인트(Policy Administration Point : PAP), 권한 검증을 수행하는 정책 결정 포인트(Policy Decision Point : PDP), 요구 및 응답 컨텍스트의 변환을 담당하는 컨텍스트 핸들러, 속성 정보의 수집을 담당하는 정책 정보 포인트(Policy Information Point : PIP), 접근 요청자와의 통신을 담당하고 접근제어와 관련된 부가적인 서비스(Obligation)를 수행하는 정책 집행 포인트(Policy Enforcement Point : PEP)로 구성된다. XACML에서 정책을 구성하는 주요 요소는 PolicySet, Policy, Rule이다. PolicySet은 정책 구성의 최대 단위로서 다수의 Policy를 포함할 수 있으며 Policy는 다수의 Rule을 포함할 수 있으며 Rule은 정책 구성의 최소 단위이다. PolicySet과 Policy는 독립적인 정책으로 사용될 수 있으나, Rule은 정책의 설정은 가능하지만 독립적인 정책으로는 사용될 수 없다. 각 요소들은 공통적으로 Target을 포함하고 있으며 Target은 접근 주체를 나타내는 Subject, 접근 대상을 나타내는 Resource, 접근 주체의 접근 대상에 대한 행위를 나타내는 Action으로 구성된다. 권한 검증의 결과는 접근 요구의 Target과 정책의 Target을 비교하고 하위 정책이 존재할 경우 하위 정책의 Target 및 조건 등을 비교하여 결정한다.

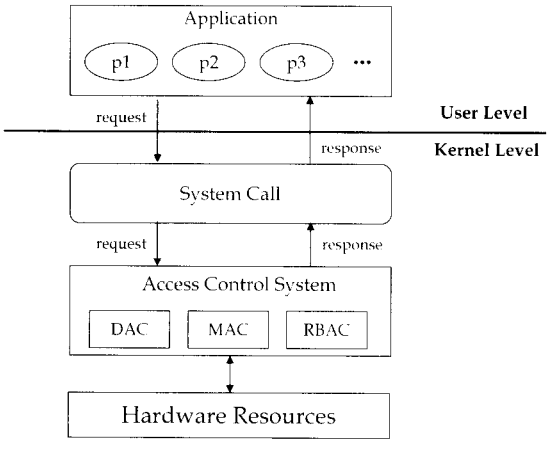


(그림 1) XACML 접근제어 모델의 데이터 플로우 다이어그램

결과 값에는 Permit, Deny, NotApplicable, Indeterminate 가 있다.

2.2 보안 운영체제(Secure OS)의 접근제어

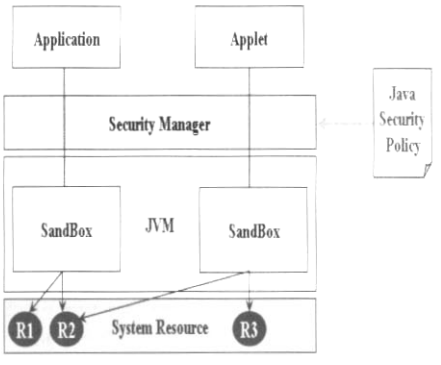
보안 운영체제는 기본적인 보안 계층을 파일 시스템, 디바이스, 프로세스 등에 대한 접근 권한 결정이 이루어지는 운영체제의 커널 레벨로 낮춘 시스템이다[11]. 이는 응용 레벨에서 일어나는 보안상의 문제점을 해결하기 위한 방법이다. 현재 공개 운영체제를 바탕으로 활발한 연구가 진행되고 있으며 대표적인 시스템으로는 Security Enhanced Linux(SELinux)와 TrustedBSD가 있다. 접근제어 방식은 (그림 2)와 같으며, 사용자 프로세스는 시스템 콜을 통해서 접근 허가를 요청한다. 접근제어 시스템의 권한 검증은 커널 모드에서 이루어지며 DAC, MAC, RBAC 등을 기반으로 정해진 접근제어 정책을 참조하여 권한 검증을 한다. 그러나 보안 운영체제는 주로 시스템 리소스에 대한 권한을 관리하기 때문에 응용 서비스에 대한 권한 정책을 설정하기가 어려우며 정책 설정이 너무 단순하여 세밀한 접근제어를 제공하기 어려운 단점이 있다.



(그림 2) 보안 운영체제의 접근제어 모델

2.3 자바2 플랫폼의 접근제어

자바2 플랫폼은 네트워크 환경을 기반으로 하여, 하드웨어 및 운영체제에 대하여 독립성을 보장함으로써 코드의 이동성에 초점이 맞추어진 플랫폼이다. 즉, 프로그래밍 언어 자체적인 보안 요소를 통하여 일정수준의 보안 서비스를 제공하고, 가상머신을 통하여 수행 코드가 하드웨어 및 운영체제에 대한 독립성을 가지게 되며, 샌드박스 모델을 통하여 자신이 수행되는 환경을 정의하는 모델을 지향한다. 바이트 코드라 불리는 중간코드를 가상머신이 실행하는 과정에서 코드 검증기가 애플리케이션의 바이트코드를 검증하여 예외를 출력하는 방식을 통하여 접근제어에 대한 보안위협으로부터 보호한다. 특히 샌드박스 모델은 (그림 3)에서와 같이 특정 자바 애플리케이션이나 애플릿이 수행되는 환경을 정책을 통하여 설정하고 자신이 수행되는 환경이외의 동작을 수행하거나 시스템 자원에 접근할 경우 자바 보안 관



(그림 3) 자바2의 접근제어

```
grant codeBase "file:${java.home}/lib/ext/*" {
        permission java.security.AllPermission;
};

// Applet security policy
grant signedBy "hylee", codeBase "155.230.106.77/download/Applet.jar" {
        permission java.io.FilePermission "${/}etc${/}hosts","read,write";
};

// Default security policy
grant {
        permission java.net.SocketPermission "localhost:1024-", "listen";
        permission java.util.PropertyPermission "java.version", "read";
        permission java.util.PropertyPermission "java.vendor", "read";
                                ...
};
```

(그림 4) 자바2의 접근제어 정책

리자가 이것에 대한 예외를 발생시킴으로써 허용된 자원과 동작만 수행할 수 있도록 한다[12].

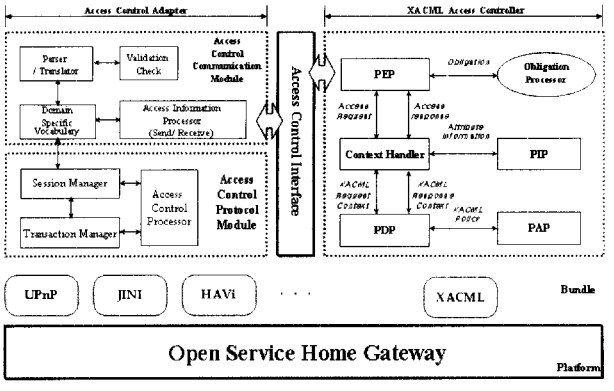
(그림 4)는 자바2 플랫폼에서 사용되는 접근제어 보안정책을 나타낸다. 이는 자바 고유의 방식으로 기술되어 다른 시스템에서 사용하는 접근제어 보안정책과의 호환 및 연동에 있어서의 문제점을 내포하고 있으며, 확장성에 있어서도 그 한계점을 드러내고 있다.

3. XACML 접근제어 시스템의 개요

본 장에서는 XACML 접근제어 시스템의 전체 구조를 설계하고 이를 구성하는 주요 모듈에 대한 기능 및 특징에 대하여 기술한다. 그리고 주요 구성 요소간의 메시지 흐름에 따른 동작 과정을 상세히 설명한다.

3.1 전체 시스템의 구조

홈 네트워크 댁내 정보가전 기기로 연결된 네트워크와 댁외의 네트워크 환경, 그리고 이들을 연결하기 위한 홈 게이트웨이로 구성된다. 특히, 홈 게이트웨이는 외부와 댁내를 연결해주는 입구로서의 역할을 하기 때문에 통합적인 보안 관리를 구축하기에 적절하다. 본 논문에서 제안하는 XACML



(그림 5) 전체 시스템의 구조

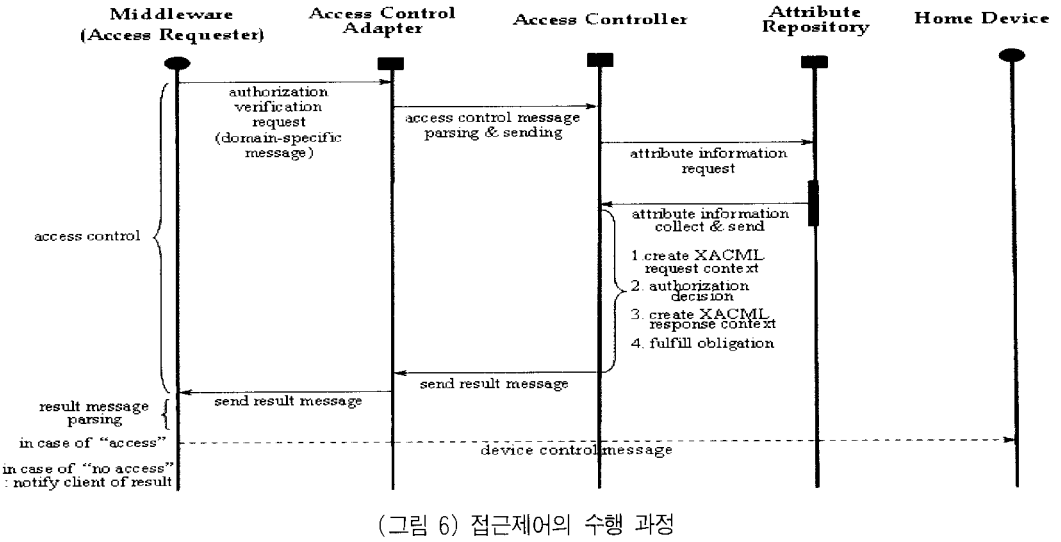
접근제어 시스템은 홈 게이트웨이에 존재하여 댁내 또는 외부에서 접근하는 사용자에 대한 접근제어를 수행하며 전체 구조는 (그림 5)와 같다. 이는 크게 접근 제어기, 접근제어 어댑터, 정책 관리기의 세 부분으로 나누어지며 주요 특징은 다음과 같다.

접근 제어기는 접근 요구 메시지와 정책을 비교하여 권한 검증을 수행하는 PDP, XACML 컨텍스트의 변환 및 역변환을 수행하는 컨텍스트 핸들러, 정보 저장소와 연결되어 속성 정보를 수집하는 PIP, 결과 정보의 처리 및 접근제어에 대한 부가적인 서비스인 Obligation 서비스를 처리하는 PEP로 구성된다. 각 요소들은 접근 제어를 위한 역할을 분담하며 순차적으로 작업이 수행되기 때문에 기능 확장이 용이하다는 장점이 있다. 접근제어 어댑터는 미들웨어와 접근제어기를 연동하기 위한 부분으로서 이는 크게 접근제어 통신 모듈과 접근제어 프로토콜 모듈로 나누어진다. 접근제어 통신 모듈은 서비스에 대한 세션을 관리하는 세션 매니저와 홈 디바이스로 전달되는 제어 명령을 전송하기 전에 권한 검증을 수행하고 그 결과에 따른 작업 및 오류를 처리하기 위한 트랜잭션 매니저로 구성된다. 접근 제어 프로토콜 모듈은 접근제어 통신 메시지를 분류하는 Domain Specific Vocabulary(DSV), 홈 디바이스로 전달되는 제어 명령으로부터 Subject, Resource, Action에 대한 정보를 추출하여 접근 요구 메시지로 변환하고 전달된 접근 응답 메시지를 파싱하는 파서, 접근 요구 및 응답 메시지를 검사하는 검증기로 구성된다. 정책 관리기는 (그림 5)에서 PAP를 나타내며 이는 XACML 접근제어 모델에 포함되는 요소이지만 접근제어기의 구성 요소는 아니다. 정책 관리기는 접근 제어기의 PDP와 연동되어 정책의 등록 및 관리 기능을 담당하며 RBAC 접근제어를 위한 Subject, Resource, Action에 대한 속성 및 계층 정보를 관리한다.

3.2 접근제어의 수행 과정

미들웨어는 홈 디바이스로 제어 명령을 전달하기 전에 접근제어를 수행한 후 결과에 따라 제어 명령을 처리한다. 미들웨어와 홈 디바이스는 기기 제어를 위한 요소이며 접근제어 어댑터, 접근제어기, 정책 관리기는 접근제어를 위한 요소이다. 접근 제어를 위한 동작 과정은 (그림 6)과 같다. 미들웨어는 접근제어 어댑터에게 권한 검증을 요구한 후 기기 제어 명령을 전달한다. 접근제어 어댑터는 접근제어를 위한 권한 정보를 생성하기 위하여 제어 명령을 파싱하여 접근 주체, 접근 대상, 접근 행위 등과 같은 권한 검증에 필요한 정보를 획득한 후, 이를 이용하여 접근 요구 메시지를 생성하고 이를 접근 제어기로 전달한다.

접근 제어기는 전달받은 접근 요구 메시지에 포함된 접근 주체, 접근 대상, 접근 행위, 접근 환경 등에 대한 속성 정보를 수집하기 위하여 속성 정보 저장소에 쿼리를 보낸다. 속성 정보 저장소는 Subject, Resource, Action 정보에 대한 속성을 저장하고 있으며 해당되는 정보를 수집한 후 접근제어기로 다시 전달한다. 속성 정보를 전달받은 접근 제어기는 권한 검증을 위해 다음과 같이 4가지 작업을 수행한다. 첫째, 접근 요구 메시지와 속성 정보를 이용하여 XACML 접근 요구 컨텍스트를 생성한다. 둘째, 생성된 XACML 접근 요구 컨텍스트의 내용에 부합하는 정책이 존재하는지를 검사하기 위해 정책 저장소에 저장되어 있는 XACML 정책



(그림 6) 접근제어의 수행 과정

을 검색한 후 권한을 검증한다. 그리고 권한 검증의 결과를 포함하는 XACML 접근 응답 메시지를 생성한다. 셋째, XACML 접근 응답 메시지를 접근제어 어댑터로 전달한다. 넷째, 적용된 정책에 Obligation 서비스가 설정되어 있을 경우 이를 수행한다. Obligation 서비스는 정책 내에 여러 개가 존재할 수 있기 때문에 서비스도 여러 개가 동시에 수행될 수 있다. 접근제어 어댑터는 최종 결과를 미들웨어에게 전달하며 미들웨어는 전달된 결과 값을 분석하여 접근이 허용되었을 경우 홈 디바이스로 제어 명령을 전송하고, 그렇지 않을 경우 접근 실패 메시지를 클라이언트에 전달한다.

4. 구 현

본 장에서는 XACML 접근제어 시스템의 구현 환경과 주요 구성 요소인 접근 제어기, 정책 관리기, 접근제어 어댑터의 구조를 설명하고 이를 바탕으로 구현한 내용에 대하여 설명한다.

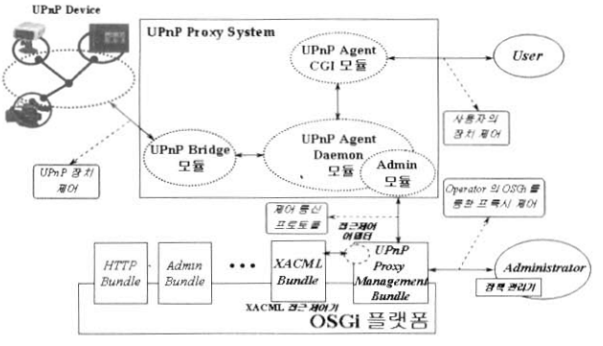
4.1 구현 환경

XACML 접근제어 시스템의 구현 환경은 (그림 7)과 같다. 홈 네트워크 환경을 구성하기 위해 OSGi 플랫폼이 존재하고 댁내 또는 댁외에서 원격으로 UPnP 디바이스를 제어하기 위한 UPnP 프락시 시스템과 이를 관리하기 위한 UPnP 프락시 관리 번들이 존재한다. UPnP 프락시 시스템은 클라이언트와의 통신 및 이벤트 처리 등을 담당하는 Agent 모듈과 디바이스와의 통신, 제어, 이벤트 처리 등을 담당하는 Bridge 모듈로 구성된다. 이러한 UPnP 프락시 시스템에 접근제어를 제공하기 위하여 XACML 접근 제어기를 OSGi의 번들로 탑재하여 UPnP 프락시 관리 번들과 연동하여 접근제어를 수행한다.

4.2 XACML 접근 제어기

4.2.1 정책 집행 포인트(PEP)

PEP는 접근 요청자인 UPnP 프락시 관리 번들로부터 전



(그림 7) 구현 환경

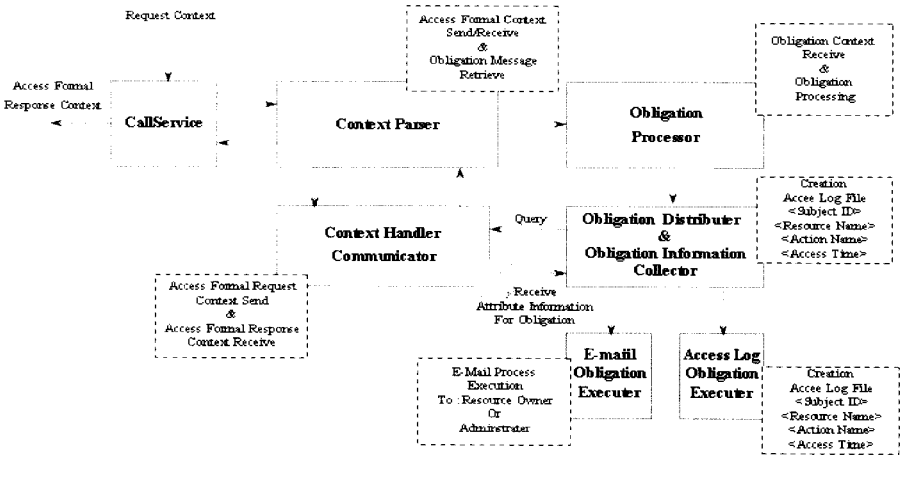
달받은 접근제어 정보를 처리하고 이에 대한 권한 검증의 결과를 전달하는 접근제어기의 에이전트 역할을 담당한다. 또한 결과에 따라 수행되는 부가 서비스인 Obligation 서비스를 처리한다. (그림 8)은 PEP의 구조를 나타낸다.

4.2.2 컨텍스트 핸들러(CH)

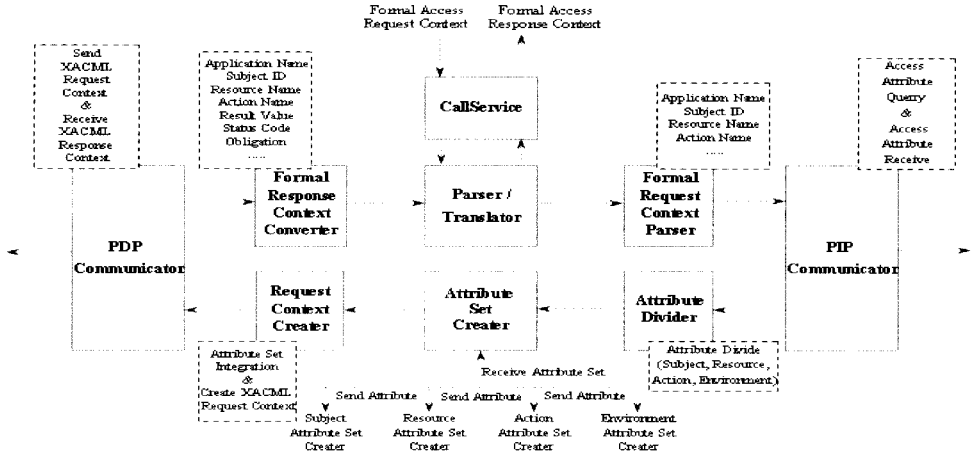
컨텍스트 핸들러는 PEP에서 전달받은 미들웨어에서 사용하는 접근 요구 메시지를 PIP에서 전달된 속성 정보를 이용하여 XACML 접근 요구 메시지로 변환하는 역할을 한다. 또한 PDP에서 전달받은 XACML 접근 응답 메시지를 미들웨어에서 사용되는 접근 응답 메시지로 변환하는 역할을 담당한다. (그림 9)는 컨텍스트 핸들러의 구조를 나타낸다.

4.2.3 정책 정보 포인트(PIP)

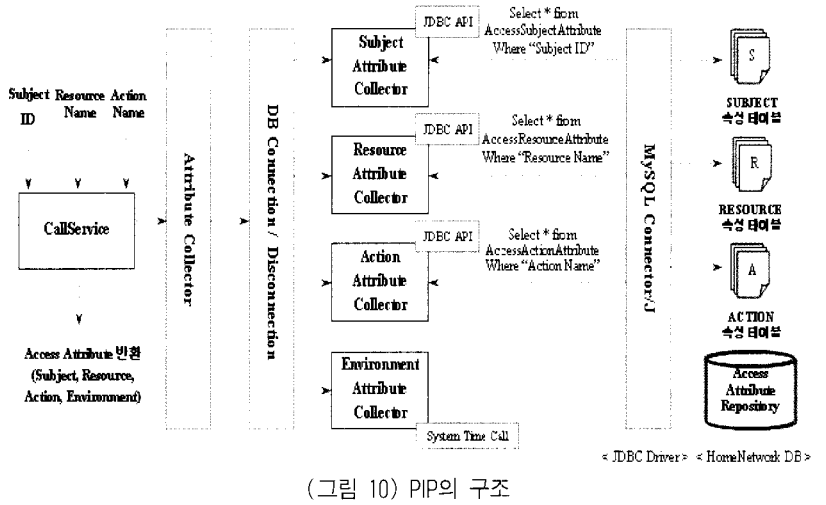
PIP는 DB에 저장되어 있는 Subject, Resource, Action, Environment에 대한 속성 정보를 수집하여 이를 컨텍스트 핸들러에 전달하는 역할을 한다. (그림 10)은 PIP의 구조를 나타내며 속성 정보를 수집하는 모듈과 DB와의 연동을 위한 모듈로 나누어진다. DB Connection 모듈은 속성 정보를 수집하기 위하여 DB에 쿼리를 보내는데, Subject의 경우 ID 속성, Resource, Action의 경우는 Name 속성을 전달한다. Environment의 경우 시스템에서 제공하는 Current Time을 이용한다.



(그림 8) PEP의 구조



(그림 9) 컨텍스트 핸들러의 구조



(그림 10) PIP의 구조

XACML에서는 속성 정보를 이용하여 RBAC 접근제어를 제공한다. 따라서 본 논문에서는 홈 네트워크 환경을 고려한 Subject, Resource, Action, Environment의 속성 정보를 정의하였다. Subject 속성은 ID, GROUP, ROLE, NAME, ISSUER, ISSUE INSTANCE로 구성되며, 댁내 사용자를 고려하여 정의하였다. Resource 속성 정보는 NAME, LEVEL, TYPE, SERVICE TYPE, OWNER Email ADRESS로 구성되며 일반적인 홈 디바이스의 특성을 고려하여 정의하였으며 Action 속성은 NAME, LEVEL, TYPE으로 구성되고 홈 디바이스 제어 명령의 특성을 고려하였다. 그리고 Environment 속성은 접근 시간을 제한하기 위한 속성이다.

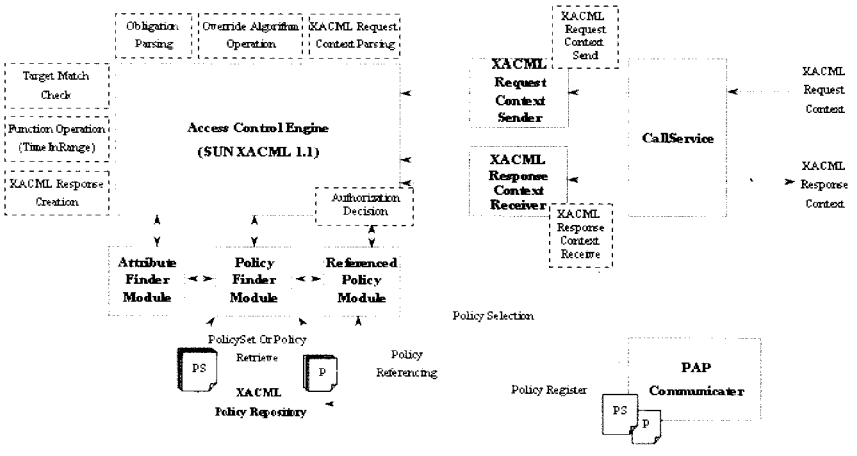
4.2.4 정책 결정 포인트(PDP)

PDP는 등록된 정책을 바탕으로 권한 검증을 수행하며 주요 모듈 및 구조는 (그림 11)과 같다. 접근제어 엔진은 SUN에서 제공하는 XACML1.1 라이브러리를 사용하였으며 정책 참조 기능 및 접근 제한 시간의 계산은 라이브러리에 포함된 클래스를 확장하여 구현하였다.

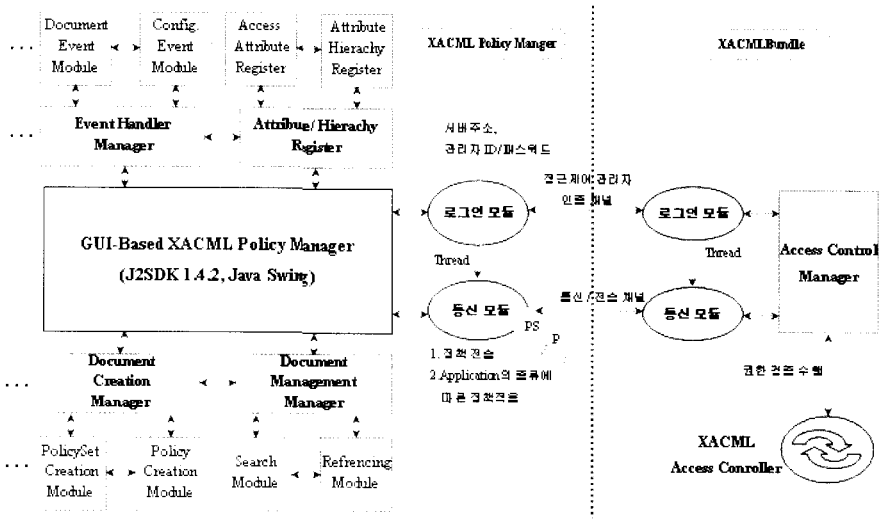
PAP에서 생성된 정책은 정책 전송 채널을 통해 PolicySet과 Policy로 구분되어 특정 디렉토리 내에 파일로 저장되며 PDP는 권한 검증을 수행할 때 이러한 정책 파일들을 등록한다. Top-Level 정책 파일은 하나만 존재할 수 있기 때문에 정책을 구성하는 단위인 PolicySet과 Policy는 중복되는 정책 내용을 가질 수 없다.

4.3 정책 관리기 (PAP)

정책 관리기의 구조는 (그림 12)와 같다. 이는 XACML 접근제어 모델에서 PAP를 나타내며 관리자를 위한 정책의 생성, 삭제, 수정, 참조 등의 기능을 제공한다. 또한 미들웨어나 애플리케이션에 따라 정책을 분류해서 관리하고 적용할 수 있으며 속성 정보의 등록 및 속성 계층의 설정 기능을 제공한다. 정책을 생성하기 위해서는 속성 계층의 설정이 필요하기 때문에 다음과 같이 속성 계층을 설정하여 정책의 권한 검증에 적용시켰다. <표 1>은 정책 관리기를 이용하여 생성한 XACML 정책이며 역할이 Administrator로 할당되어 있는 사용자는 모든 리소스에 대하여 모든 제어가 가능하다는 것을 나타낸다. Rule 부분에는 더 자세한 조건이 설정되어진다.



(그림 11) PDP의 구조



(그림 12) 정책 관리기의 구조

〈표 1〉 XACML 접근제어 정책

```
<?xml version="1.0" encoding="euc-kr"?>
  <Policy>
    <Target>
      <Subjects>
        <Subject>
    <SubjectMatch MatchId="string-equal">
      <AttributeValue DataType="string">Administrator</AttributeValue>
      <SubjectAttributeDesignator AttributeId="role"
       DataType="string"/>
    </SubjectMatch>
        </Subject>
      </Subjects>
      <Resources>
        <AnyResource/>
      </Resources>
      <Actions>
        </AnyAction>
      </Actions>
    </Target>
    <Rule RuleId="1" Effect="Permit">
                ... ... ...
    </Rule>
  </Policy>
```

4.4 접근제어 어댑터

접근 제어 어댑터는 XACML 접근 제어 번들, UPnP 프락시 시스템, UPnP 프락시 관리 번들을 연동하기 위한 모듈로서 접근 제어를 위한 메시지를 전달하거나 처리하는 역할을 담당한다. 이를 위해 UPnP 프락시 시스템의 Agent 모듈에 접근 요구 및 결과 처리를 위한 Access Control Function이 존재하고, UPnP 프락시 관리 번들에 접근제어 번들과의 연동 및 통신 메시지의 파싱을 위한 모듈이 존재한다. UPnP 프락시에서 UPnP 프락시 관리 번들로 전달되는 접근 요구 메시지의 주요 정보는 userId, deviceId, actionName이며 이는 XACML 접근 요구 메시지의 Subject ID, Resource Name, Action Name과 동일하다.

5. 실험 및 결과

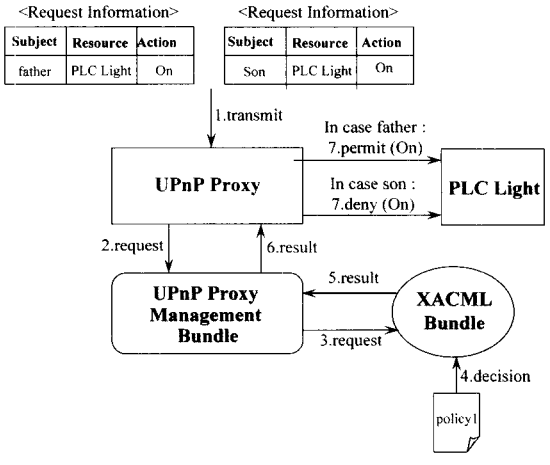
본 장에서는 XACML 접근제어 시스템을 OSGi기반 UPnP 프락시 시스템에 적용하여 접근제어 관련한 실험한 내용에 대하여 설명하고 결과를 분석한다. 실험 시나리오는 3가지로 구성되며 다양한 정책에 대하여 접근제어가 수행되는지를 검증하고 XACML이 제공하는 정책 참조 및 시간에 따른 접근제어 기능을 테스트한다. 첫 번째 실험 시나리오는 아버지의 경우 댁내 PLC 전구를 제어하기 위한 모든 권한을 가지는 것이며, 두 번째 실험 시나리오는 아버지는 댁내 오디오를 제어하기 위한 모든 권한을 가지지만 어머니는 시작과 정지만 할 수 있다. 마지막으로 세 번째 실험 시나리오에서는 딸은 허가된 시간 내에서만 오디오에 대하여 모든 권한을 가지는 것을 보여준다. 실험을 위한 시스템의 수행 과정은 다음과 같다. 첫째, UPnP 프락시 관리기를 이용하여 OSGi의 UPnP 프락시 관리 번들 및 XACML 접근제어 번들을 활성화시킨다. 둘째, 접근제어 관리기를 이용하여 접근제어 정책을 생성한다. 셋째, 서비스 화면을 이용하여 서비스를 수행한다.

5.1 적용되는 정책의 존재 유무에 따른 접근제어

첫 번째 실험은 접근 요구에 부합되는 내용을 가지는 정책이 존재하는지를 검사하여 정책이 존재할 경우 접근이 허용되고 그렇지 않을 경우 접근이 허용되지 않음을 확인한다. 정책을 설정하기 전에 접근제어를 적용하려는 미들웨어명을 UPNP로 설정하고 생성할 정책의 단위를 PolicySet으로 선택한다. 그런 후 정책 관리기를 통하여 다음과 같은 정책을 생성한다.

(정책 1) father은 PLC_Light에 대하여 모든 권한을 가진다.

father은 소속 그룹을 administrator로 설정하고 PLC_Light는 Level을 High로 설정하며 Action은 AnyAction으로 설정한다. RuleCombining 알고리즘은 deny-override로 하며, Obligation 서비스는 Permit일 경우 Access Log와 E-Mail을 선택한다. Configuration 및 정책의 설정 과정이 끝난 후



(그림 13) 정책 1을 적용한 실험

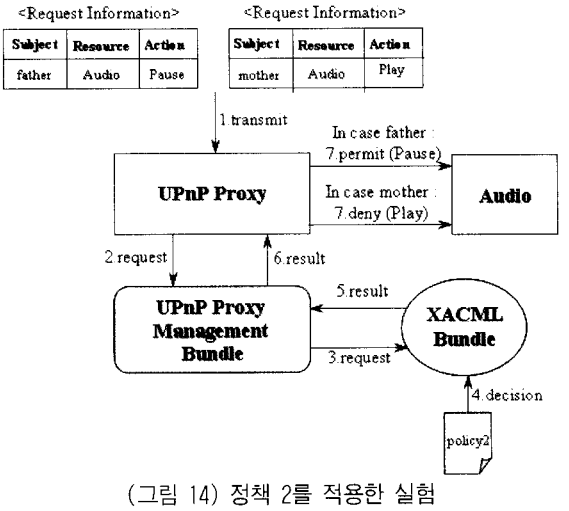
웹 브라우저의 로그인 화면을 통해 father ID로 UPnP 프락시 시스템에 로그인 한 후 PLC_Light를 제어하였고 그 결과 모든 동작이 수행됨을 확인하였다. 또한 Obligation 서비스의 수행으로 인하여 로그 파일이 생성되고 관리자에게 접근 내용 및 결과에 대한 E-mail이 전송되었다. father이 아닌 다른 ID로 로그인 한 후 제어하였을 경우에는 어떠한 동작도 수행되지 않음을 확인하였다. (그림 13)은 정책 1을 적용한 접근제어 동작 과정을 나타낸다.

5.2 Policy Reference를 적용한 접근제어

두 번째 실험은 하나의 정책을 생성하고 여기에 새로운 정책을 추가할 경우에 접근 제어가 수행되는지를 검증한다. 정책 파일의 추가는 새로운 Policy 정책 파일을 생성한 후 PolicyReference를 사용하여 PolicySet 정책 파일이 Policy 정책 파일을 참조하는 방법을 사용한다. 이는 PolicySet의 하위에 Policy가 추가된 것과 같은 효과를 가진다. 실험할 정책의 내용은 다음과 같다.

(정책 2) father는 Audio에 대하여 모든 권한을 가진다.

첫 번째 실험에서 father은 PLC_Light에만 모든 제어가

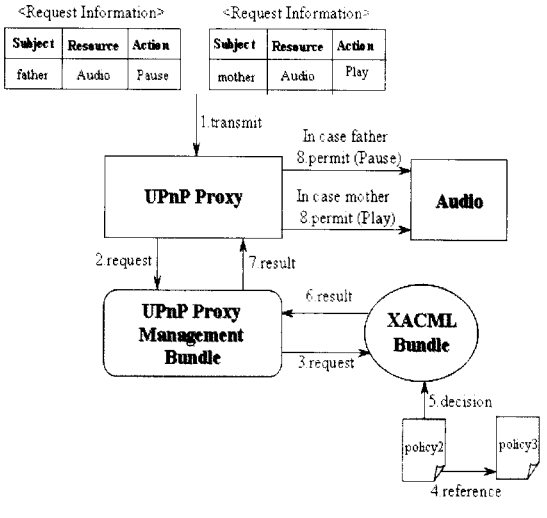


(그림 14) 정책 2를 적용한 실험

가능하였지만 <정책 2>가 생성되면 Audio에 대한 모든 제어도 가능해진다. 그러나 mother ID로 로그인 한 후 audio를 제어한다면 동작이 수행되지 않는다. 이는 mother에 대한 정책이 존재하지 않기 때문이다. (그림 14)는 정책2를 적용한 접근제어 동작과정을 나타낸다.

(정책 3) mother는 Audio에 대하여 "PLAY"와 "STOP"만을 할 수 있다.

<정책 3>은 mother이 audio에 대한 PLAY와 STOP의 제어를 가능하게 한다. 이는 정책 관리기의 Policy Reference 기능을 통하여 <정책 2>의 하위에 <정책 3>이 추가되는 효과를 가지며 audio에 대한 어떠한 권한도 가지지 않았던 mother은 Play와 Stop을 제어할 수 있는 권한을 가진다. 따라서 mother로 로그인 한 후 Audio에 대하여 Play와 Stop을 제어하면 동작이 수행되지만 다른 명령에 대해서는 동작이 수행되지 않음을 확인할 수 있다. (그림 15)는 정책2가 정책 3을 참조한 접근제어 동작과정을 나타낸다.



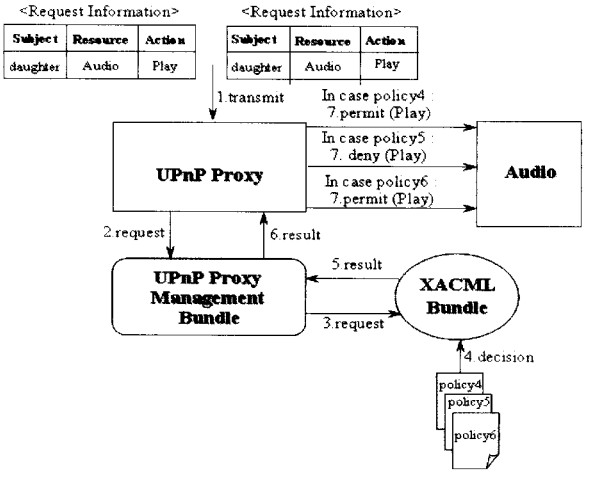
(그림 15) 정책 2가 정책 3을 참조한 실험

5.3 Condition을 적용한 접근제어

세 번째 실험은 XACML에서 제공하는 Condition을 이용하여 특정 시간에 대한 접근 제어가 수행되는지를 검증한다. 이를 위해 3단계 과정을 통하여 실험을 수행하였고 실험할 정책의 내용은 다음과 같다.

(정책 4) daughter는 Audio에 대하여 모든 권한을 가진다.
(정책 5) daughter는 Audio에 대하여 A.M 01:00부터 P.M 06:00에만 모든 권한을 가진다.
(정책 6) daughter는 Audio에 대하여 xx:xx부터 yy:yy에만 모든 권한을 가진다. (실험 수행 시간으로 설정)

<정책 4>는 daughter ID로 로그인 한 후 Audio를 제어하면 모든 제어의 수행이 가능함을 의미한다. 그러나 <정책 5>가 설정되면 시간에 따라서 사용자의 접근이 제한된다. 즉, daughter는 오전 1시부터 오후 6시에만 Audio에 대한 모든 제어가 가능하다. 만약 오후 9시에 daughter로 로그인 한 후 Audio를 제어했을 경우 어떠한 제어도 수행되지 않는다. <정책 6>은 현재 실험을 수행하는 시간을 접근허용 시간의 범위에 포함시킨 정책으로서 <정책 5>에서 동작되지 않는 Audio가 <정책 6>에 의해서 동작됨을 확인하였다. (그림 16)은 정책 4,5,6을 적용한 접근제어 동작과정을 나타낸다.



(그림 16) 정책 4, 정책 5, 정책 6을 적용한 실험

6. 향후 계획 및 결론

본 논문에서는 홈 네트워크 환경에서 접근제어 서비스를 제공하기 위하여 차세대 접근제어 표준인 XACML을 이용한 접근제어 방안을 제시하고 이를 바탕으로 접근제어 시스템을 구현하였다. 그리고 이를 OSGi기반 UPnP 프락시 시스템에 적용시켜 실험을 수행하였고 사용자의 접근 권한 및 정책에 따라 홈 디바이스에 대한 접근 제어가 수행됨을 검증하였다. 본 논문에서 구현한 XACML 접근제어 시스템의 장점은 다음과 같다. 첫째, XML기반의 접근제어 표준 기술인 XACML을 사용하였기 때문에 다양한 접근제어 기능을 제공하며 정책 설정 및 확장이 용이하다. 또한 접근제어 표준인 XACML을 사용하여 접근제어를 수행하는 서비스와의 연동이 가능하기 때문에 향후의 접근제어 기술로서 활용 가능성이 높다. 둘째, 홈 게이트웨이에서 접근제어 관리를 하기 때문에 통합적인 접근제어가 가능하다. 셋째, RBAC 및 다양한 정책 기법을 지원하기 때문에 체계적이고 복잡하지 않은 접근제어 정책을 설정할 수 있다. 넷째, 특정 미들웨어에 종속되지 않는 구조를 가진다. 이러한 특징은 기존의 홈 네트워크 접근제어 기술이 가졌던 다양한 문제점들을 해결할 수 있다.

홈 네트워크가 활성화되면 모든 정보 및 기술이 융합되는 유비쿼터스 환경으로 발전하기 때문에 기존의 홈 네트워크 환경에서의 접근제어 기술도 유비쿼터스 환경에 적합한 형

태가 되어야 한다. 유비쿼터스 환경에서는 많은 디바이스 및 센서가 존재하기 때문에 권한 체계 및 프로토콜이 간단해야 하며, 새로운 서비스를 수용할 수 있는 기술 및 유비쿼터스 환경에 최적화된 접근제어 정책 언어가 필요하다. 또한 사용자 인증 및 디바이스 인증 기술과 연동될 수 있어야 하며 다양한 미들웨어가 제공하는 서비스 환경을 수용할 수 있어야 한다. 따라서 본 논문에서 제안한 XACML 접근제어 시스템은 이러한 요구 사항을 반영한다면 향후 홈 네트워크 및 유비쿼터스 환경에서의 통합적인 접근제어 모델을 위한 좋은 참조가 될 것이다.

참 고 문 헌

[1] A. Herzog, N. Shahmehri, A. Bednarski, I. Chisalita , U. Nordqvist, L. Saldamli, D. Szentiványi, M. Östring, "Security Issue in E-Home Network and Software Infrastructures," Proceedings of the 3rd Conference on Computer Science and Systems Engineering in Linköping. Norrköping, Sweden. Pages: 155-161. Linköpings universitet. 2001.

[2] G. Steven and Ungar, "Home Network Security,"Proceedings of 2002 IEEE 4th International Workshop on Network appliances, pp.41-48, January 15 16, 2002.

[3] UPnP Forum, Understanding UPnP, June 2000, http://www.upnp.org/download/UPNP_UnderstandingUPNP.doc

[4] Sun Microsystems, Jini Architecture Specification, December 2001, http://www.jini.org/

[5] Sony, Specification of the Home Audio/Video Interoperability Architecture Version 1.0, January 2000, http://www.havi.org

[6] OSGi, Open Service Gateway Initiative 1.0, May 2000, http://www.osgi.org/

[7] UPnP Forum, Device Security and Security Console v1.0, Nomember 2003, http://www.upnp.org/standardizeddcps/security.asp

[8] Dae-Ha Park. Doo-Kwon Baik, "OSSEM: a security model for OSGi service framework," SCI 2003. 7th World Multiconference on Systemics, Cybernetics and Informatics Proceedings. IIIS. Part Vol.11, 2003, pp.189-94 Vol.11. Orlando, FL, USA

[9] OASIS, XACML 1.0 Specification, February 2003, http://www.oasis-open.org/committees/download.php/2406/oasis-xacml-1.0.pdf

[10] OASIS, SAML 1.1 Specification, 2 September 2003, http://www.oasis-open.org/committees/download.php/3406/oasis-sstc saml-core 1.1.pdf

[11] NSA, "Security Enhanced Linux," http://www.nsa.gov/selinux/

[12] Anne Anderson, "Java Access Control Mechanisms," Technical report, Sun Microsystems, March 2002, http://lists.oasis open.org/archives/xacml/200201/pdf00000.pdf

이 준 호

e-mail : jhlee7@tta.or.kr

2002년 경북대학교 컴퓨터과학과(학사)

2004년 경북대학교 대학원 컴퓨터과학과 (이학석사)

2005년~2006년 한국전자통신연구원 정보보호연구단 연구원

2006년~현재 한국정보통신협회 소프트웨어시험인증팀 전임연구원

관심분야 : 무선 인터넷, 임베디드 시스템, 홈 네트워크 보안

임 경 식

e-mail : kslim@knu.ac.kr

1982년 경북대학교 전자공학과(학사)

1985년 한국과학기술원 전산학과 (공학석사)

1994년 University of Florida 대학원 전산학과(공학박사)

1995~1998년 한국전자통신연구원 책임연구원, 실장

1998년~현재 경북대학교 전자전기컴퓨터학부 부교수

관심분야 : 이동컴퓨팅, 무선인터넷, 홈 네트워킹, 컴퓨터통신

원 유 재

e-mail : yjwon@kisa.or.kr

1985년 충남대학교 계산통계학과(학사)

1987년 충남대학교 대학원 전산학전공 (이학석사)

1998년 충남대학교 대학원 전산학 (이학박사)

1987년~2001년 한국전자통신연구원 책임연구원, 팀장

2001년~2004년 (주)안랩유비웨어 기술이사

2004년~현재 한국정보보호진흥원 IT 기반보호단 응용기술팀장

관심분야 : 네트워크 보안, 무선 인터넷, 홈 네트워크