OSGi 서비스 플랫폼에서 RBAC 기반의 사용자 접근제어 프레임워크

(A RBAC-based Access Control Framework in OSGi Service Platform)

조 은 애 † 문 창 주 †† 백 두 권 †††
(Eun-Ae Cho) (Chang-Joo Moon) (Doo-Kwon Baik)

요 약 최근 네트워크의 발전에 따라 홈 네트워크에도 많은 연구가 진행되고 있다. 현재 홈 네트워크에서는 각 디바이스별로 ACL을 이용하여 접근 제어 정책을 관리하는 방법이 가장 많이 사용되고 있고, EAM(Extranet access management)이 솔루션 형태로 응용되고 있으며, 보안 운영체제에 대한 연구가 공개 운영체제를 바탕으로 진행되고 있다. 또한 홈 서버 중심의 홈 네트워크 사용자 인증 메커니즘 등도 연구되고 있다. 그러나 이러한 연구들은 다음과 같은 문제점이 있다. 첫째, 홈 네트워크에서 사용될 것으로 예상되는 접속기술의 전송범위가 넓어 인증되지 않은 외부 단말기의 접속이 가능하고, 둘째, 사용자가 별도의 필요 정보를 일일이 디바이스에 설정할 필요가 있어 불편하다. 셋째, 사용자 프라이버시나 편의성을 고려하고 있지 않다.

이질적인 다양한 기술들이 존재하는 홈 네트워크 환경에서 OSGi는 상호운용성을 보장하는 서비스 플랫폼을 제공한다. 여기에서 사용자 접근제어는 위와 같은 문제점이 있어서는 안되는 홈 네트워크 보안의 핵심 분야 중에 하나지만 아직 구체적인 연구가 진행되고 있지 않다. 따라서 본 논문에서는 OSGi 서비스 플랫폼이 운영되는 홈 네트워크 환경에서 사용자 접근제어를 위한 RBAC 기반의 권한부여 정책 관리 프레임워크와 접근제어 운영방법을 제안한다. 제시된 접근제어 프레임워크는 OSGi 표준에서 명확하게 언급되지 않았던 부분들을 고려사항으로 나열하고, 이 문제들을 해결하는 방법들을 프레임워크로 제안한다. 제안하는 프레임워크에서는 홈 게이트웨이의 제한된 자원에 대해 RBAC의 개념을 이용하여 사용자 접근제어에 대한 정책변경의 횟수를 줄이는 효율적이고 경제적인 운영 방법을 제시한다. 또한 본 논문에서 제안하는 정책은 사용자역할정책과 권한할당정책을 개별적으로 정의한 후 사용자 정보를 댁내에서 결정함으로써, 사용자 편의성을 높여주고 프라이버시 문제를 해결하도록 방안을 제시한다.

키워드 : 접근제어, 권한부여, RBAC, OSGi

Abstract Recently, according to the network environment, there are many researches for home network. Nowadays, in home network, the method that access control policy is managed for each home device by using ACL is popular, and EAM (Extranet access management) is applied as a solution. In addition, the research about secure OS is ongoing based on open operating system and the research of user authentication mechanisms for home network using home server is also in progress. However, these researches have some problems as follows; First, the transmission scope of expected access technology in home network is wide, so unauthenticated outside terminal can access the home network. Second, user is inconvenient because user need to set the necessary information for each device. Third, user privacy and convenience are not considered.

OSGi provides a service platform for heterogeneous technologies in home network environment. Here, user access control is one of the core parts which should have no problems such as above items, but there are no concrete researches yet. Thus in this paper, we propose an access control policy management framework and access control operation based on RBAC for user access control in home network environment in which OSGi service platform is operated. First, we list the consideration

· 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성지원 사업의 연구결과로 수행되었음
† 학생회원 : 고려대학교 컴퓨터학과
eacho@korea.ac.kr
†† 정 회 원 : 건국대학교 항공우주공학 교수
cjmoon@konkuk.ac.kr
††† 종신회원 : 고려대학교 컴퓨터학과 교수
baikdk@korea.ac.kr
논문접수 : 2007년 2월 16일
심사완료 : 2007년 7월 3일

which is not clearly mentioned in OSGi standard, and then we solve these above problems through new framework. In addition, we propose the effective and economical operation method which reduces the policy change frequency for user access control by using RBAC concept though limited resource of home gateway. Besides, in this paper, these proposed policies are defined separately as user-role assignment policy and permission-role assignment policy, and user decide their own policies. In conclusion, we provide the scheme to enhance the user convenience and to solve the privacy problem.

Key words : Access Control, Authorization, RBAC, OSGi

1. 서 론

인터넷은 매우 널리 사용되고 있고, 매우 빠르게 발전하고 있다. 그리고 다양한 네트워크들이 더 효율적으로 사용되기 위해 구축되었다. 최근에는 네트워크 환경이 심지어 댁내까지 구축되어 홈 네트워크 분야가 새로운 기회를 가지게 되었다. 이 중 홈 게이트웨이는 홈 네트워크에 있는 댁내 네트워크(residential network)와 유무선 접근 네트워크(wire/wireless access network)를 연결함으로써 많은 서비스들을 제공한다. 뿐만 아니라 휴대용 장치들을 통한 원격 자동 제어와 홈 보안기능 및 관리기능도 제공한다. 따라서 향후 홈 게이트웨이는 홈 네트워크가 구축되어 있는 디지털 홈의 핵심 역할을 수행할 것으로 예상된다.

홈 네트워크 환경에는 많은 유무선 네트워크 기술들을 기반으로 다양한 홈 게이트웨이, 지능형 정보가전 및 미들웨어들이 존재한다. 또한, 서로 다른 하드웨어 플랫폼, 운영체제 및 네트워크 프로토콜에 따라 수많은 서비스 개발 환경이 존재한다. 이와 같은 홈 네트워크의 이질성과 복잡성을 해결하기 위하여 OSGi(Open Service Gateway initiative)는 동일한 형태의 API(Application Programming Interface) 형식을 통하여 서비스 제공자, 네트워크 관리업체, 시스템 개발 회사들 및 정보가전 기기 업체 간의 상호운용성을 보장하는 서비스 플랫폼을 정의하고 있다[1]. 따라서 홈 네트워크에서의 보안 문제는 홈 게이트웨이 상에서의 OSGi 서비스 플랫폼과 밀접한 관계를 가지고 있다고 할 수 있다.

홈 게이트웨이 상에서의 주요 핵심 보안 이슈는 홈 네트워크에 접근하는 사용자에 대한 인증(Authentication)과 접근제어(Authorization)이다[2]. 인증은 신뢰할 수 있는 제 3의 기관에 의해서 실행되고 기존의 인증 기술들이 이미 성숙해 있어 상당 부분 홈 게이트웨이와 독립적으로 수행된다[1,3]. 그러나 접근제어는 홈 게이트웨이와 매우 밀접한 관련이 있다. 사용자가 홈 게이트웨이에서 댁내 정보가전과 자동화기기에 접근할 때마다 그것을 접근할 수 있는지의 여부를 판단하는 작업이 이루어져야 하기 때문이다.

또한 홈 네트워크에 대한 많은 연구가 현재 진행되고 있다. 홈 네트워크 접근 제어는 각 디바이스별로 ACL을 이용하여 관리 정책을 사용하고 있으며, 특히 접근 제어와 관련해서는 EAM(Extranet access management) [4]이 솔루션 형태로 응용되고 있다. EAM은 싱글사인온(Single Sign-On: SSO)과 사용자 역할 기반의 세분화된 접근 관리(granular access control)를 제공하는 전략적인 솔루션으로 사용자의 인증을 관리하고 애플리케이션이나 데이타에 대한 사용자 접근을 결정하는 비즈니스 정책을 구현한다. 국내는 소프트포럼, 이니텍 등이, 국외는 네티그리티(Netegrity)가 시장을 주도하고 있다. 또한 보안 운영체제[5]는 기본적인 보안 계층을 파일 시스템, 디바이스, 프로세스 등에 대한 접근 권한 결정이 이루어지는 운영체제의 커널 레벨로 낮춘 시스템이며, 시장은 컴퓨터어소시에이트, 시만텍 등이 주도하고 있다[6,7]. 뿐만 아니라 홈 서버 중심의 홈 네트워크와 관련하여 '홈서버 중심의 홈네트워크 사용자 인증 메커니즘[8]'이 현재 국내 표준으로 제정되어 있는 상태이다. 그러나 관리자가 각 가정에 한 명씩 존재하여 사용자 등록, 디바이스 등록, 접근제어 관리, 서비스 관리 등을 하도록 한다는 내용만 있을 뿐 아직 홈 서버 기반의 구체적인 접근 제어가 제시되어 있지 않은 상태이다. 위에서 나열한 연구들은 다음과 같은 문제점들을 가진다. 첫째, 홈 네트워크에서 사용될 것으로 예상되는 접속기술의 전송범위가 넓어 인증되지 않은 외부 단말기의 접속이 가능하다. 둘째, 사용자가 별도의 필요 정보를 일일이 디바이스에 설정할 필요가 있어 불편하다[9]. 셋째, 사용자 프라이버시나 편의성을 고려하고 있지 않다.

본 논문의 기반이 되는 OSGi 서비스 플랫폼은 가장 최근에 배포된 버전 4.0의 서비스 명세 부분에서 접근제어 부분을 사용자 관리 서비스 부분에 언급하고 있다[1]. 그러나 여기에 언급된 내용은 표준이 가지고 있는 한계점으로 인하여 몇 가지 문제점을 찾을 수 있다. 첫째, 위에서 언급된 중요 내용들인 새로운 보안 취약성과 프라이버시 문제에 대해 구체적으로 설명하지 못하고 있다. 둘째, 실제적인 OSGi 기반의 홈 네트워크 서비스를 위하여 필요한 주체, 대상 등이 명확한 접근제어 프레임워크가 제시되지 못하고 있다. 셋째, 사용자의 편의성에 대한 부분은 고려되지 않고 있다. 결국, OSGi는 서비스를 위한 미들웨어로 번들의 설치 및 서비스 참조

에 대한 보안 기술은 제공되지만 서비스를 이용하는 사용자에 대한 보안 기술은 제공하지 않고 있는 것이다[10]. 이를 해결하기 위하여, 홈 네트워크에서의 보안 기술, 특히 접근 제어의 필요성은 다음과 같이 나열할 수 있다[6,7].

- 인터넷과의 연결로 기존 해킹 공격기술이 그대로 적용될 수 있으므로 다양한 위험으로부터 홈 네트워크 자원을 보호해야 한다.
- 생체정보 등을 기반으로 한 상황인지 홈 서비스의 증가로 프라이버시 침해가능성이 더욱 증가한다.
- 홈 네트워크 환경에 적합한 접근제어 기술이 필요하다.
- 시스템의 성능과 더불어 접근 제어의 활용에 용이성과 관리성을 고려해야만 한다.

먼저, 홈 네트워크는 기존의 보안 취약성을 안고 가는 동시에 프라이버시 문제 등의 신규 보안 취약성이 발생한다. 다양한 유무선 네트워크 및 프로토콜로 구성되어 있기 때문에 그에 따라 다양한 보안취약성이 발생하고, 상대적으로 저성능인 홈 디바이스 사용증가로 해킹 공격대상의 가능성이 높아진다. 네트워크 상에서 인증되지 않은 오퍼레이터에 의해 악의적인 서비스가 배치되거나 서비스가 변질 될 위험이 생길 수도 있다. 프라이버시 침해와 관련해서는 지금까지 홈 네트워크 보안기술은 방범/방재 분야를 중심으로 개발되어 왔으며, 최근 일부 업체에서만 관련 IT 보안 기술을 개발하였다. 뿐만 아니라 홈 네트워크 보안기술은 홈 게이트웨이 중심의 가상사설망(Virtual Private Network: VPN), 방화벽(Firewall) 등이 대부분이므로 더욱 견고한 프라이버시 보호가 필요하다. 마지막으로, 홈 디바이스의 특성 및 사양을 고려하여 홈 디바이스에 주는 영향을 줄일 수 있는 경량화 된 기술이 필요하고, 사업자가 가입자 홈 디바이스에 대한 접근제어 권한을 보유하는 문제에 대한 고려가 필요하다. 누가 관리를 담당하게 되는지에 관계없이 접근 제어 관리의 편의성이 고려되어야만 하는 것이다.

서비스 플랫폼과 관련하여, OSGi 기반의 인프라스트럭처로는 온톨로지, 상황 인식 애플리케이션에 대한 연구[11-15]가 진행되고 있다. 홈 네트워크워크를 위한 온톨로지 지식서비스 모델은 PersonRelative, Entity-Relative, Space, Time, Session, UserActivity, Event, Policy Ontology 등의 요소로 구성되며, 상황인지 기반 지능적 홈 네트워크 서비스를 위한 기반 기술로는 리즈너(Reasoner) 기술, 마이너(Miner) 기술, 온톨로지(ontology) 구성 기술 등을 필요로 한다. 그러나 이러한 연구들에서 조차 RDF(Resource Description Framework)[16], OWL(Ontology Web Language)[17] 등의 언어로 온톨로지를 구축할 때 사용자가 온톨로지 표준에 따라 직접 구축하기 쉽지 않고, 그 보안 및 프라이버시 역시 보장할 수가 없기 때문에 앞에서 언급한 문제점에 대해서는 해결 방안이 되지 못한다. 덧붙여 OSGi 서비스 프레임워크는 특성상 오퍼레이터가 많은 사용자와 서비스 번들을 관리해야 하기 때문에 접근제어에 많은 자원과 노력이 필요하므로 효과적인 접근제어 모델, 접근제어 정책 작성 및 관리, 접근제어 프레임워크 운영 방법 등이 필요하다.

따라서 본 논문에서는 이러한 문제들을 해결하기 위해 OSGi 서비스 플랫폼이 운영되는 홈 네트워크 환경에서 사용자 접근제어를 위한 RBAC(Role-Based Access Control)[18-20]기반의 사용자 접근제어 프레임워크와 접근제어 운영방법을 제안한다. 먼저, 제시된 접근제어 프레임워크는 OSGi 표준에서 명확하게 언급되지 않았던 부분들을 고려사항으로 나열하고, 이들을 해결하는 방법들을 프레임워크로 제안한다. 사용자의 댁내 정보가전과 자동화 기기들의 접근권한을 명세한 접근제어 정책을 작성하고 관리하는 것은 개인 프라이버시와 밀접한 관련이 있으므로 이를 보호할 수 있는 접근 제어 프레임워크를 제시한다. 또한, 접근제어 정책과 관련하여 현재 명확하게 제시되어 있지 않은 정책 생성과 관리에 대해 주체와 대상, 저장 장소 등을 효율적으로 구성하고 운영할 수 있도록 한다. 뿐만 아니라, 댁내 사용자들의 편의성을 최대한 보장하면서 프라이버시가 보장되는 접근제어 방법을 제시한다. 만일 정책 작성이나 권한부여의 과정에서 프라이버시를 고려하지 않은 기관이나 업체가 관리의 목적으로 권한부여 정보를 소유하면 댁내 구성원이나 정보가전 관련 정보들이 외부에 노출되고, 반대로 프라이버시 보장을 위하여 사용자의 편의성을 무시하면 사용자들로부터 배척될 수 있기 때문이다. 더불어 홈 게이트웨이의 제한된 자원을 이용하여 사용자 접근제어를 효율적이고 경제적으로 운영할 수 있는 방법을 제시한다. 이러한 프레임워크는 홈 네트워크에서 실제 사용자 접근제어를 하는데 효과적인 가이드라인을 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 연구배경에 대하여 언급하고 3장에서는 본 논문에서 제안하는 접근제어 프레임워크의 전체적인 구조에 대하여 설명한다. 4장에서는 정책의 구성과 그 예에 대해서 설명하고 5장에서는 제안된 접근제어 프레임워크 운영에 대하여 언급한다. 6장에서는 본 논문에서 제안된 사용자 접근제어 방법과 기존의 방법에 대하여 정량적인 부분과 정성적인 부분을 비교 평가한다. 마지막으로 7장에서는 본 연구의 결론 및 향후 연구 과제에 대해 서술한다.

2. 연구 배경

2.1 OSGi 서비스 플랫폼

OSGi는 홈 게이트웨이 표준화 단체 중의 하나로 업계표준을 정하는 단체이다. 현재 OSGi에는 썬 마이크로시스템즈, IBM, 소니, 삼성 등 세계 유수의 기업이 참여하고 있다. 서비스에 대한 기본적인 프레임워크는 거의 완성되었지만, 보안에 대해서는 아직도 많은 부분이 부족하다. OSGi 서비스 플랫폼의 전체적인 구조는 그림 1과 같다[21].

OSGi 서비스 플랫폼은 서비스 제공자(Service Provider), 오퍼레이터(Operator), 홈 게이트웨이(Home Gateway)로 구성된다[2,22]. 서비스 제공자는 서비스 번들을 제작하여 오퍼레이터에게 제공한다. 오퍼레이터는 각 가정의 홈 게이트웨이를 관리하는 사업자로 홈 게이트웨이에 대한 정보, 사용자 정보, 번들 관련 정보들을 관리한다. 홈 게이트웨이는 댁내 망과 외부망의 경계선에 위치하며 여러 번들과 사용자 정보 등을 가진다. OSGi 프레임워크는 자바 프로그래밍 언어의 플랫폼 독립성과 동적 코드 로딩 능력을 이용하여 소형 메모리 디바이스에 적합한 응용프로그램인 번들을 쉽게 개발하고 동적으로 배치할 수 있도록 한다[22]. 홈 게이트웨이에 배치된 여러 번들은 사용자가 원하는 서비스를 제공하며 사용자의 필요에 따라 오퍼레이터로부터 동적으로 설치 제거된다. 사용자는 댁내 사용자와 외부 사용자로 구분된다. 댁내 사용자는 하나의 홈 게이트웨이에서 여러 서비스를 사용하는 사람이고, 외부 사용자는 각 홈 게이트웨이를 돌아다니면서 특정 서비스를 사용한다[2].

2.2 RBAC 모델

RBAC 모델은 1970년대에 개척된 온라인 시스템의 개념으로 다중 사용자, 다중 애플리케이션과 함께 시작되었다[2,18]. RBAC 모델은 역할(role)의 개념을 사용함으로써 사용자와 그들의 권한(permission)들을 효과적으로 관리할 수 있다. RBAC에서 권한은 역할과 관련이 있다. 사용자들은 역할의 한 멤버가 됨으로써 사용자의 권한을 가진다. 이 기본 개념은 권한의 이해와 관리를 간편하게 해주는 장점이 있다. 그림 2는 RBAC 모델의 개념적인 그림이다[2,20].

이 모델은 기본적으로 사용자(users: U), 역할(roles: R), 권한(permissions: P), 세션(sessions: S)의 4개의 요소를 포함한다. 사용자(U)는 인간의 행동이나 자율적인 에이전트를 나타낸다[2]. 반면에 역할(R)은 조직에서 역할에 속해 있는 멤버들에게 주어진 권한과 책임감을 고려하여 연관된 의미들을 가진 일의 기능이나 이름이다. 권한(P)는 시스템에서 하나 이상의 대상(object)에 대한 접근의 특정 형태에 대한 허가이다. 그림 2는 또한 사용자 할당(user assignment: UA)과 권한 할당(permission assignment: PA)이 다-대-다 관계를 가지고 있는 것을 보여준다. 제약조건(constraints)은 UA와 PA에서 관찰되어야만 하는 규칙을 서술한다. 역할 계층(Role hierarchy: RH)은 역할의 계층적인 구조와 제약조건의 특정한 형식을 나타낸다.

2.3 접근제어 구조

많은 애플리케이션과 사용자들이 존재하는 대규모의 시스템에서 권한의 관리는 매우 어려운 작업이다. 이러한 문제들을 해결하기 위한 접근제어 관리 솔루션들은 모두 중앙에서 권한부여 정보를 관리하는 방법을 사용하고 있다. 권한부여 정보를 중앙 집중적으로 관리하는 방법에는 그림 3의 user-pull 구조와 그림 4의 server-pull 구조가 있다[23].

그림 3은 user-pull 구조로 사용자는 권한부여 서버를 통해서 인증 받고 애플리케이션에 접근하기 위한 일

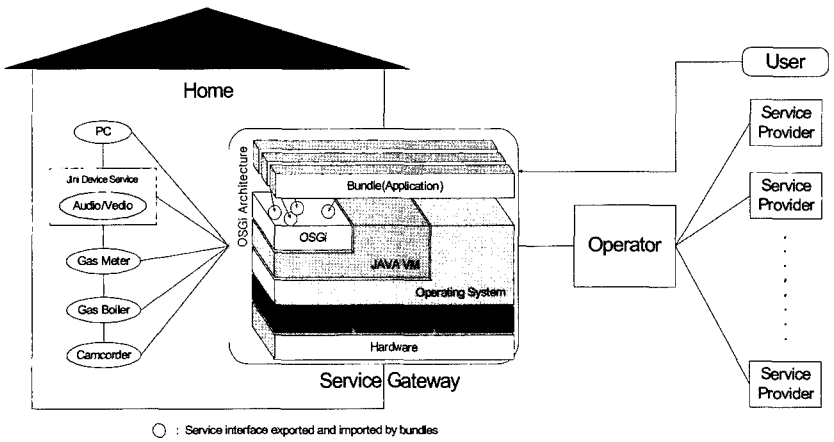

그림 1 OSGi 서비스 플랫폼

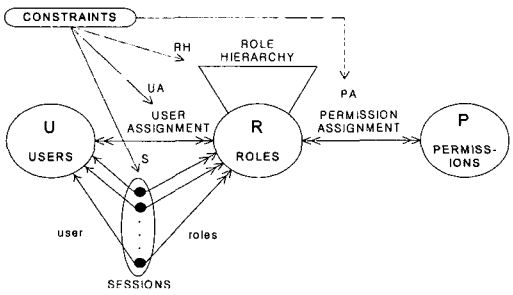

그림 2 RBAC 모델

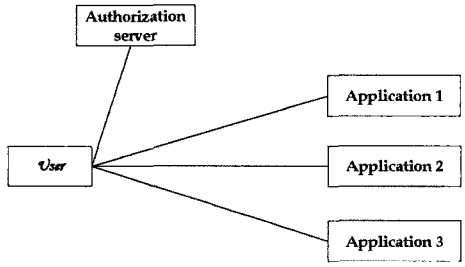

그림 3 User-pull 구조

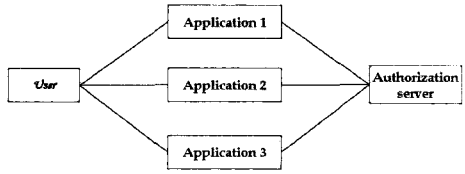

그림 4 Server-pull 구조

종의 증명서를 받는다. 그 다음 애플리케이션에 접근하기위한 권한부여 시 증명서를 보여준다. 그림 4는 server-pull 구조로 애플리케이션이 사용자를 인증하는 것이 요구되고, 권한부여 서버에 사용자의 권한에 관한 정보가 모아진다. 사용자가 애플리케이션에 접근하기를 시도할 때마다, 애플리케이션은 사용자의 권한을 결정하기 위해서 권한부여 서버에 질의한다[20,24].

3. 접근제어 프레임워크

이 장에서는 홈 네트워크에서 사용자 접근제어를 위하여 고려해야 하는 사항을 4가지 측면에서 분석한다. 이 분석 내용을 기반으로 OSGi 서비스 플랫폼에 가장 적합한 사용자 접근제어 프레임워크를 제안한다.

3.1 고려사항

홈 네트워크에서의 사용자 접근제어 프레임워크에 대한 고려사항은 접근제어 모델, 권한부여 구조, 접근제어 정책 작성 및 관리, 접근제어 정책 배치 등의 관점에서 나열한다.

고려사항 1. 접근 제어 모델

가능한 접근제어 모델의 종류에는 DAC(Discretionary Access Control), MAC(Mandatory Access Control), RBAC 등이 있다.

① DAC : 정보의 소유자(생성자)가 자신의 정보에 대한 접근을 통제한다. 즉, 정보 소유자의 재량에 따라 접근제어가 이루어지는 모델이다. DAC에 의한 접근통제는 주로 접근통제목록(Access Control Lists: ACLs)을 사용하여 이루어진다. ACL의 등록사항(entry)을 간단하게 추가, 변경 혹은 삭제함으로써 접근 대상(Object)에 대한 접근통제를 손쉽게 할 수 있다[20]. 그러나 정보 소유자가 모든 권한을 가지고 있으므로, 그 외의 사람들의 프라이버시나 의지 등은 반영되기 힘들다. 또한 확장성이 부족하여 사용자나 접근대상이 많아지면 ACL을 관리하는 것이 어렵다.

② MAC : 자동적으로 수행되는 정보흐름 규칙을 기반으로 한다. 주체(Subject)와 객체(Object)에 대한 레벨을 구성하여 낮은 레벨의 주체가 높은 레벨의 객체에 접근할 수 없도록 하는 모델이다[23]. 객체의 소유자가 아닌 관리자만이 정보자원의 분류를 설정 및 변경하고 접근권한을 부여함으로 DAC 보다는 안전하다. 기밀성을 중요시하는 환경에는 적합하나 융통성이 부족하여 일반적인 비즈니스 환경에 사용하기는 적합하지 않다[23,25]. 데이타 소유자의 결정을 무효로 만들 수 있으므로 역시 프라이버시를 침해당할 수 있다.

③ RBAC : 2.2절에서 언급한 것과 같이 사용자와 권한 사이에 역할이 있다. 중앙의 관리자는 주체와 객체와의 관계를 설정하여 접근을 결정한다. 즉, 사용자의 역할에 근거하여 자원에 대한 접근을 허용한다. 사용자와 권한 관리에 대한 경제성과 효율성이 높은 모델로 사용자와 접근 대상이 많은 엔터프라이즈 환경에 적합하다. 또한 뛰어난 확장성을 가지고 있어 사용자나 권한이 추가되는 경우 역할을 이용하여 쉽게 대처할 수 있다. 그러나 관리자가 많은 부담을 가지고 있고 중앙관리를 위한 많은 비용이 소요된다.

고려사항 2. 권한부여 구조[26,27]

앞의 2.3에서 언급한 바와 같이 운영구조는 user-pull, server-pull의 2가지 접근 방법이 있다.

① User-Pull 구조 : 사용자가 일단 자신의 인증 정보를 받으면 그것의 유효 값이 만료될 때까지는 사용할 수 있다. 따라서 서비스를 이용할 때 권한부여 서버에 매번 접속할 필요가 없으므로 서버의 부담을 감소시킨다. 그러나 사용자가 서버로부터 인증을 받는 것이 먼저 필요하고, 증명서를 IT전문가가 아닌 일반인들이 관리해야하므로 사용자의 세심한 주의가

요구된다. 게다가, 사용자 장치나 애플리케이션이 가지고 있는 정책에서 사용자 혹은 서비스에 대한 권한을 매번 검사해야만 한다. 이것은 자원이나 리소스에 한계가 있는 홈 게이트웨이와 정보가전에서 처리하기에 부담이 갈 수 있으며, 자원을 활용하기 위한 프로그래머들의 구현 로드가 커진다. 뿐만 아니라 메커니즘이 변경될 때마다 각각의 홈 게이트웨이에 대해서 다시 수정해주어야 하며, 유효 기간이 있기 때문에 일단 증명서를 받으면 중간에 메커니즘이 변경되어도 사용자에 대한 업데이트가 느려질 수 있다.

② Server-Pull 구조 : 사용자가 애플리케이션을 사용하고자 할 때, 애플리케이션은 사용자 권한부여를 결정하기 위해서 인증 서버에게 질의한다. User-pull 구조와는 달리, 사용자가 서비스를 사용하기 전에 서버에 접속해서 접근 제어 정보를 가져올 필요가 없으므로 사용자 편의성을 제공한다. 사용자는 인증에 대한 부담 없이 애플리케이션에 바로 접근할 수 있다. 그러나 사용자가 서비스를 원할 때마다 장치들이 서버에 접속해야만 하기 때문에 서버의 측면에서는 약간의 부담이 생길 수 있다.

고려사항 3. 접근제어 정책 작성 및 관리

접근제어 정책을 작성할 가능성이 있는 사람은 크게 댁내 사용자, 오퍼레이터, 서비스 제공자로 나눌 수 있다.

① 댁내 사용자(Residential User) : 사용자가 모든 정책을 작성하는 경우, 사용자 프라이버시가 보장될 수 있고 동시에 홈 네트워크는 제 3자의 간섭 없이 운영될 수 있다. 그러나 사용자는 IT전문가가 아니기 때문에 다양한 장치와 서비스에 대한 정책을 작성하기 위해서 많은 시간을 들여야 할 필요가 있다. 게다가 사용자는 정책 작성과 관리에 대한 능력과 의지가 없다. 사용자는 정책 작성에 대한 기본 지식을 갖추고 있지 않기 때문에 오류의 가능성도 있다.

② 오퍼레이터(Operator) : 오퍼레이터가 정책을 관리할 경우, 사용자는 정책을 직접 작성할 필요가 없다. 그래서 정책 작성의 오류를 감소시킬 수 있고 사용자는 정책 작성의 시간을 절약할 수 있다. 오퍼레이터는 전문가들이기 때문에 정책 관리에 대한 부분은 믿을 수 있다. 그러나 실제적으로 사용자의 의도에 따라 정책을 작성하는 것이 불가능하다. 따라서 프라이버시 문제가 발생할 수 있고, 정책 관리가 모든 가정의 경우를 고려해야하기 때문에 각 사용자에게 맞는 관리가 어렵다.

③ 서비스 제공자(Service Provider) : 이 경우에 정책은 권한과 그 내용을 잘 알고 있는 서비스 제공자에 의해 만들어진다. 그래서 정책은 합리적으로 생성될 수 있다. 반면, 많은 서비스 제공자들은 자신의 이익에 따라 정책을 만들게 된다. 그러므로 정책을 만들기 전에 오퍼레이터 혹은 공신력 있는 기관이 정책의 형식을 정의해야만 한다. 그리고 정책 통합과 호환 문제도 정의되고 해결되어야 한다.

고려사항 4. 접근제어 정책 배치

접근제어 정책이 저장되는 장소는 정책의 관리에 많은 영향을 미친다. 정책은 권한부여 서버, 각 가정에 있는 홈 게이트웨이 혹은 사용자가 사용하는 각각의 디바이스에 저장될 수 있다.

① 권한부여 서버 : 정책을 권한부여 서버에 배치하면, 각 가정의 홈 게이트웨이에서 관리하는 것보다 더 전문적이고 체계적인 관리를 기대할 수 있다. 그러나 이 경우에는 사용자가 서비스를 이용하기 위해 어떤 권한을 확인할 때마다 권한부여 서버에 접속해야만 한다. 그렇기 때문에 네트워크 트래픽이 유발된다.

② 각 가정의 홈 게이트웨이 : 사용자 혹은 장치가 서버로부터 정보를 가져와야 할 필요가 없기 때문에 사용자가 서비스를 이용하기를 원할 때 빠른 권한부여가 가능하다. 그리고 서버가 한 번에 많은 가정의 정책들을 관리할 때 일어나는 네트워크 트래픽 문제가 생기지 않는다. 그러나 서비스 번들이 바뀌거나 업데이트될 때 홈 게이트웨이는 정책을 변경해야하는 부담이 생긴다. 그리고 일반 사용자가 정책을 작성할 경우, 모든 정책을 관리하기 때문에 관리상의 허점이 생길 수 있다.

③ 각 가정의 홈 장치들 : 제품의 특성에 알맞은 정책을 작성하고 적용하는 것이 가능하다. 그러나 정책의 수정, 삽입, 삭제와 같은 조작이 필요한 경우, 각 장치마다 일일이 적용되어야 하므로 관리 시 많은 작업이 필요하다. 또한 저사양 레벨 장치의 경우 정책의 관리 및 운영에 어려움이 있다.

3.2 프레임워크

접근 제어는 인증된 사용자의 접근을 허용할 것인지의 여부를 결정하는 과정이다. 3.1의 고려사항을 기반으로 하여 본 논문에서는 OSGi 서비스 플랫폼에서 효율적인 접근제어 프레임워크를 제안한다.

접근 제어에서 접근을 하는 실체의 집합을 '주체'(Subject)라고 하고, 수동적인 자원의 집합을 '객체'(Object)라고 한다[23]. 따라서 홈 네트워크에서는 접근의 주체가 댁내 사용자 혹은 믿을 수 있는 외부의 사용자가 되며, 접근의 객체는 정보가전을 사용하거나 조절하기 위해 서비스 제공자가 공급한 서비스 번들의 서비스들이 해당된다. 권한(Permission)은 사용자가 특정 서비스 번들의 서비스에 접근할 수 있는 권한을 의미한다. 서비스 번들이 정보가전들을 컨트롤하기 때문에 홈 게이트웨이의 서비스 번들에 대한 접근여부는 궁극적으

로 사용자가 특정 정보가전에 접근할 수 있는지를 결정하게 된다.

다음은 3.1절의 고려사항을 기반으로 OSGi 환경에 접합한 프레임워크 구성요소들을 도출한 결과이다.

적용 사항 1. 접근 제어 모델

접근 제어 모델은 OSGi 프레임워크의 특성상 RBAC 모델을 선택하였다. MAC은 사용자가 자신의 권한을 제어할 수 없다. DAC에서 ACL을 이용하여 권한을 제어할 수 있는 방법이 있지만, IT전문가가 아닌 일반 사용자가 ACL을 작성하고 운영하는 데에는 문제가 있다. OSGi에서는 많은 수의 사용자와 다양한 종류의 정보가전, 그리고 이에 따른 서비스 번들들이 존재한다. 수많은 세대에서 변경이 일어나므로 많은 세대들을 한꺼번에 관리하기 위해서는 확장이 어려운 ACL보다는 RBAC과 같은 융통성이 있고 경제성이 있는 모델이 적합하다. 더 나아가 제안한 프레임워크는 여러 세대들을 관리하는 것에서 비즈니스 모델을 창출할 수도 있다.

적용 사항 2. 권한부여 구조

홈 네트워크 환경은 주거 환경과 매우 밀접해서 사용자 편의성이 매우 중요하다. 운영 구조 중에서 user-pull은 서비스가 추가되거나 정책이 변경되었을 때 그 적용 속도가 느리다. 반면에 server-pull 구조는 정책의 업데이트 속도가 빠르고 사용자 편의에 좋다[22]. 그러나 네트워크 트래픽을 고려할 때 잦은 권한부여 정보에 대한 질의는 서버에게 부담이 되어 서버의 효율성을 감소시킨다. 따라서 본 논문에서는 변형된 server-pull 방식을 이용하여 서버 효율성 문제를 개선한다. 이 방식은 사용자가 서비스를 요청했을 때 사용자가 직접 서버에서 권한할당 여부를 확인하지 않는다. 대신에 제안한 구조와 같이 홈 게이트웨이의 권한 정보(Permission-role Information)에서 접근 가능 여부를 확인할 수 있도록 하였다[28].

적용 사항 3. 접근 제어 정책 작성 및 관리

어느 한 곳에서 모든 정책을 작성하는 것은 프라이버시와 사용자 편의성 측면에서 적합하지 않다. 가정의 프라이버시에 관련된 정보를 전적으로 외부에 의존하는 것은 바람직하지 않기 때문에 사용자는 자신의 집에 해당하는 권한부여 정책을 작성하고 관리하는 것이 필요하다. 그러나 사용자들은 권한부여 정책을 작성하고 관리할 능력이 없으며 이것은 사용자 편의성 측면에서도 좋지 않아 사용자들로부터 외면당할 수 있다. 즉, 사용자 편의성을 중요한 요소로 꼽고 있는 홈 네트워크 환경에는 적합하지 못한 측면이 있다. 따라서 본 논문에서는 먼저 권한부여 정책을 사용자할당(user-role assignment)정책과 권한 할당(permission-role assignment) 정책의 두 부분으로 나눈 다음 그것을 각각 작성한다. 권한 할당 정책은 오퍼레이터가 작성하고, 사용자할당정책은 사용자가 작성하도록 하였다. 전문가인 오퍼레이터는 각각의 서비스들을 분석하고 역할들을 정의한 다음 그 역할에 알맞은 권한을 할당한다. 사용자는 그 역할들을 검토한 후 사용자 할당을 작성한다. 이렇게 하면 사용자는 사용자의 의지대로 역할을 결정할 수 있고 제 3자가 역할 할당에 참여하거나 관여할 수 없기 때문에 프라이버시도 보장할 수 있다. 게다가 사용자는 역할 할당을 제외한 다른 정책들을 만들지 않기 때문에 정책 작성에 있어 사용자 편의성을 제공해준다.

적용 사항 4. 접근제어 정책 배치

사용자할당정책은 홈 게이트웨이에서 관리하고, 권한할당 정책은 권한부여 서버로부터 홈 게이트웨이에 다운받는다. 이때, RBAC 서버에 사용자할당정책을 보관할지 여부는 프라이버시를 위해서 매우 중요한 문제이다. 사용자할당정책은 사용자가 작성하지만 외부에 이 정책들의 저장과 관리를 맡기게 되면 또다시 프라이버시 문제와 마주치게 되기 때문이다. 그래서 본 논문은 각각의 홈 게이트웨이에서 작성된 사용자 할당 정책의 정보를 암호화하여 RBAC 서버에 저장하는 방법을 제안한다. 이렇게 하면, 권한부여 서버에 의한 체계적인 백업과 기술적인 관리도 기대할 수 있다. 또 RBAC 서버와 홈 게이트웨이 사이의 데이타 교류도 쉬워지므로 정책변경과 애플리케이션 업데이트가 느리다는 관리상의 단점을 해결할 수 있다.

위의 내용을 기반으로 하여 본 논문에서는 그림 5의 접근제어 프레임워크를 제안한다.

각각의 홈 게이트웨이는 모든 가정에 분산되어 있다고 가정한다. 홈 네트워크 환경에서의 시스템 운용은 미들웨어인 JINI, UPnP(Universal Pulg and Play)를 모두 사용할 수 있고, Java 기반의 API를 제공해주고 있는 OSGi 서비스 플랫폼을 중심으로 네트워크 환경을 제공하며, 대부분의 미들웨어들은 기기 간의 통신을 위한 TCP/IP 프로토콜을 하부에 사용하고 있다. 먼저, 댁내 사용자는 로그인 번들(Login Bundle)을 통하여 인증서버에 접근하고 인증을 받는다. 즉, 인증 번들은 인증서버와 연동하여 ID/PW, 인증서 등의 이미 잘 알려져 있는 다양한 방법을 통하여 인증을 제공한다. 인증된 사용자가 번들(Bundle) 접근을 통하여 특정 정보가전에 대한 서비스 사용을 요청하면 OSGi 프레임워크의 권한부여 검사(Authorization Check) 기능은 XML 형태의 권한부여 정책(Authorization Policy)을 체크하여 사용자가 번들 서비스에 접근이 가능한지 여부를 판단한다. 권한부여 서버인 RBAC 서버는 사용자들에 대한 권한부여 정보(Authorization Information)를 관리한다. 각 홈 게이트웨이는 RBAC 서버의 권한부여 정보 중 해당

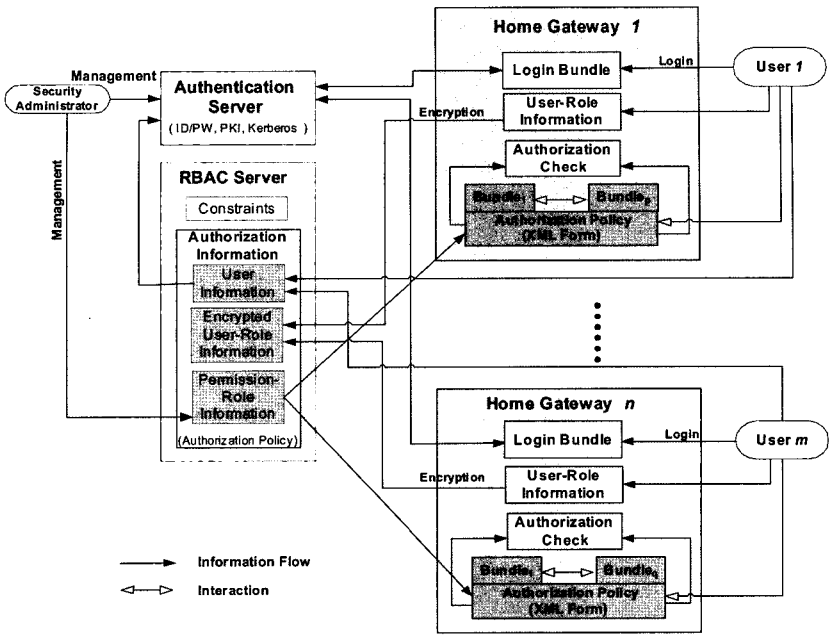

그림 5 제안된 접근 제어 프레임워크

가정에 대한 권한 정보(Permission-Role Information)만을 관리한다. 권한할당 정보는 홈 게이트웨이에 배치되는 권한부여 정책(Authorization Policy)의 실제 내용이 된다. 또한 이 정보들은 앞에서 언급한 바와 같이 체계적인 관리와 백업을 위해서 암호화되어 RBAC 서버에 저장된다. 보안 관리자는 인증 서버와 RBAC 서버를 관리한다. 인증 번들은 별도의 인증 서버와 함께 홈 게이트웨이에서 인증 서비스를 제공한다. 만약 권한부여 정책에 사용자에 대한 특정한 번들의 접근 권한이 명세되어 있다면, 사용자는 번들로부터 서비스를 제공받을 수 있다.

이 제안된 권한부여 구조는 앞에서 언급한 바와 같이 RBAC 모델에서 server-pull을 변형하여 적용한 것이다. 사용자가 서비스를 요청했을 때, 사용자는 권한부여 할당 정보를 홈 게이트웨이에서 검사할 수 있다. 즉, 권한부여 정보가 RBAC 서버뿐 아니라 위의 그림과 같이 홈 게이트웨이에도 있는 것이다. 이것이 기존의 server-pull 구조와는 다르다. 또한 사용자는 '적용사항 3'과 같이 일일이 디바이스에 있는 접근 제어 리스트를 변경하거나 요청할 필요 없이 댁내 사용자에 대한 사용자역할 정보만 설정하면 되는 것이 기존의 ACL 기반의 방법과 다른 점이며 이 부분에서 사용자 편의성을 만족시킬 수 있다. 제안한 방법에서 정책을 관리하는 장소는 홈 게이트웨이와 오퍼레이터 두 군데로 나누어진다. 앞에서 언급한 바와 같이 사용자정책은 홈 게이트웨이에서 관리되고 권한부여정책은 오퍼레이터에서 관리되는 것이다. 따라서 서비스를 제공받기 위해서 사용자가 요청을 했을 때, 서비스가 제공되기까지의 과정이 추가가 되므로, 서비스 제공 시간이 상대적으로 길어질 수 있다.

이를 해결하고자, 본 논문의 제안 방법에서는 각 가정에 해당하는 권한부여정책을 오퍼레이터에서 받는 것이 아니라 홈 게이트웨이에 다운받아 사용함으로써 서비스 지연 시간을 줄이고자 하였다. 이렇게 하면, 권한부여정책의 업데이트가 있을 경우에만 홈 게이트웨이가 다운받으면 되므로 서비스를 이용하기 위해 권한부여정책에 접근하는 시간은 단축시킬 수 있고, 사용자가 어떠한 디바이스를 사용하기 위해 권한부여정책에 접근하는지를 알리지 않아도 되기 때문에 홈 네트워크 내의 프라이버시를 보장할 수도 있다.

4. 권한부여 정책

홈 게이트웨이에 있는 권한부여 정책들은 XML 형식으로 데이타베이스에 있는 권한 테이블 정보의 일부분으로 표현된다. 권한부여 인터페이스에서 역할을 획득하는 메소드는 인증된 사용자에게 할당된 역할을 파악하고 조사한다. 서비스를 확인하는 메소드는 XML을 기반으로 하여 권한부여 정책에서 역할에 대한 서비스 권한을 확인하게 된다. 만약 역할이 권한을 가지고 있다면 사용자는 요청된 서비스에 접근할 수 있다. 만약 가지고 있지 않으면 사용자는 그것에 접근할 수 없다. 권한부여

```
<?xml version="1.0" encoding="EUC-KR"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="authorization_policy">
        <xsd:complexType>
            <xsd:element name="role" minOccurs="1" maxOccurs="unbounded">
                <xsd:complexType>
                    <xsd:element name="bundle" minOccurs="1" maxOccurs="unbounded">
                        <xsd:complexType>
                            <element name="service  type="xsd:string" minOccurs="1"
                                maxOccurs="unbounded"/>
                            <xsd:attribute name="b_name" type="xsd:string" use="required" />
                        </xsd:complexType>
                    </xsd:element>
                    <xsd:attribute name="r_name" type="xsd:string" use="required"/>
                </xsd:complexType>
            </xsd:element>
            <xsd:attribute name="g_ip" type="xsd:string" use="required" />
        </xsd:complexType>
    </element>
</xsd:schema>
```

그림 6 권한부여 정책의 권한할당부분 스키마의 예

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="user_info">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="name" type="xsd:string" minOccurs="1"/>
                <xsd:element name="role" type="xsd:string" minOccurs="1"/>
                <xsd:element name="access identity" type="xsd:string" minOccurs="1"/>
                <xsd:element name="clearance" type="xsd:string" minOccurs="1" />
            </xsd:sequence>
            <xsd:attribute name=registry_num type="xsd:string" use="required">
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

그림 7 사용자할당 부분 스키마의 예

정책에 접근하기 위해서 SAX(Simple API for XML) [29]나 DOM(Document Object Model)[30] 등을 사용할 수 있다.

RBAC의 개념을 이용한 사용자 접근제어 정책은 앞에서 언급한 바와 같이 크게 사용자역할정책과 권한부여 정책으로 나눌 수 있다. 정책의 작성 시 요소로 사용자 정보, 역할 정보, 홈 게이트웨이 정보, 댁내에서 서비스를 제공해주는 번들과 장치 정보, 권한부여 정보 등이 필요하다. 그림 6과 7에서 나타내고 있는 XML 스키마와 같이 권한부여 정책은 역할(role), 번들(bundle), 각 번들에서 제공하고 있는 서비스들(service), 홈 게이트웨이 주소(g_ip)로 구성되어 있으며, 사용자역할정책은 사용자 이름(name), 사용자의 역할(role), 사용자의 ID (access identity), 기밀취급 허가(clearance)로 구성되어 있다. 그림 6은 권한부여 정책 스키마의 한 예를 나타내고, 그림 7은 사용자 할당 스키마의 한 예를 나타낸다.

그림 6은 역할이 다양한 번들을 가질 수 있다는 것을 정의하고 각 번들은 적어도 하나 이상의 서비스를 제공할 수 있다. 그림 7에서 사용자는 identity, role, clearance 등의 엘리먼트들을 가진다.

그림 8은 그림 6을 사용하여 XML 형식으로 나타낸 권한부여 정책의 예를 보인다. 홈 게이트웨이는 163.152.41.173이고 사용자는 Adult, Child, Inspector 등의 역할

```
<?xml version="1.0" encoding="EUC-KR"?>
<authorization-policy  g_ip = "163.152.41.173">
    <role  r_name = "Adult">
        <bundle  b_name = "VideoService">
            <service> MovieOrder </service>
            <service> AdultMovieOrder </service>
        </bundle>
        <bundle  b_name = "WatchCamera">
            <service> MonitorService </service>
            <service> ControlService </service>
        </bundle>
    </role>
    <role  r_name = "Child">
        <bundle  b_name = "VideoService">
            <service> MovieOrder </service>
        </bundle>
        <bundle  b_name = "WatchCamera">
            <service> MonitorService </service>
        </bundle>
    </role>
    <role  r_name = "Inspector">
        <bundle  b_name = "GaugeExamination">
            <service> ElectricExamService </service>
            <service> GasExamService </service>
            <service> TapWaterExamService </service>
        </bundle>
    </role>
</authorization-policy>
```

그림 8 권한부여 정책 XML의 예

```
<?xml version="1.0" encoding="EUC-KR"?>
<user_info registry_num = "98765432">
    <name>Robert Cho</name>
    <role> Adult </role>
    <access_identity> Bob </access_identity>
    <clearance> Approval </clearance>
</user-info>
```

그림 9 사용자 할당 정책 XML의 예

을 정할 수 있다. 각각의 역할은 번들과 사용자에 따라 알맞은 서비스를 가진다. Adult의 경우, Common Movie와 Adult Movie들을 VideoService 번들을 통해서 주문할 수 있다. 그리고 Monitor와 Control 권한은 WatchCamera 번들에서 요청될 수 있다. 반면에 Child의 경우에는 Adult Movie 주문 서비스나 Monitor Control 서비스가 허가되지 않는다. Inspector의 경우, 이것이 공공 서비스를 관리하는 것과 관련된 역할이므로, 검침 서비스 번들에서 전기, 가스, 수도 검침 등을 요청할 수 있다.

그림 9는 그림 7에 따른 사용자 할당 XML의 한 예이다. 이 그림에서 사용자는 'Robert Cho'라는 사람이고 등록 번호는 '98765432'이다. Robert는 Adult 역할을 가지고, ID로 Bob을 쓰고 있다. 이 사용자는 또한 Clearance가 허용되어 있으므로 기밀 사항에 접근할 수 있다.

5. 접근제어 프레임워크 운영

5.1 프레임워크 운영

댁내에는 적은 수의 사용자와 제한된 자원으로 홈 게이트웨이를 운영하게 된다. 만일 사용자가 직접 댁내에

서 접근제어 정책을 작성하고 관리한다면 한정된 댁내의 자원을 운영하기에 부담감이 존재한다. 왜냐하면 RBAC은 대규모 환경에 적합한 모델이기 때문이다. 오히려 사용자와 서비스를 직접 맵핑하는 ACL(Access Control List)과 같은 접근제어 방법이 적합할 수도 있다[23]. 그러나 앞의 3, 4절에서 언급한 것과 같이 사용자가 직접 정책을 전부 작성하고 관리하는 것이 프라이버시는 보호할 수 있지만, 정책 작성의 효율성, 정책관리의 전문성, 사용의 편리성, 정책의 이동성 측면에서 결코 합리적이지 않다. 그러므로 본 논문에서 제안한 것과 같이 오퍼레이터에 의한 관리가 필요하다. 그리고 많은 사용자와 다양한 정보가전, 자동화 기기들에 대한 서비스들을 운영해야하게 되므로 범위는 댁내와 비교할 수 없을 만큼 넓어지게 된다. 따라서 RBAC 모델을 사용하는 것이 적합하다는 결론을 얻을 수 있다.

본 논문에서 제안한 정책들은 홈 네트워크의 홈 게이트웨이에서 사용자역할정책이 적용될 수 있도록 애플리케이션이 갖추어져 있어야 하며, 권한부여정책 역시 오퍼레이터가 각 홈 게이트웨이에 분배하기 위해서 분산된 엔터프라이즈 환경에 알맞은 J2EE와 같은 서버 환경이 구축되어 있어야 한다. 위와 같은 환경에서 RBAC 모델을 적용하기 위해, 프레임워크 운영 방법을 홈 게이트웨이(Home Gateway), 오퍼레이터(Operator), 서비스 제공자(Service Provider)를 중심으로 그림 10과 같이 나타냈다.

Step 0 : 서비스 제공자는 서비스를 개발한다.

Step 1 : 서비스 제공자는 서비스 등록기(Service Registry)에 개발한 서비스 번들을 등록한다.

Step 1-1 : 서비스 제공자는 등록한 서비스에 대한 권한 정보를 권한부여 서버에 분배한다.

Step 2 : 보안 관리자는 각 서비스에 대한 정보를 검색한다.

Step 3 : 보안 관리자는 권한부여 서버에 권한부여 정책을 업데이트 한다.

Step 4 : 권한부여 서버는 서비스 번들을 배치하기 전에 업데이트된 권한부여 정책을 받아서 새로운 정책과 기존의 정책 간의 충돌을 검사한다.

Step 5 : 사용자는 역할 형식과 권한을 검색한다.

Step 6 : 사용자는 검색된 내용을 참조하여 해당하는 역할을 할당한다.

Step 7 : 사용자는 홈 게이트웨이에 작성된 사용자 할당 정책을 저장한다.

Step 8 : 권한부여 서버는 권한 할당 정보를 홈 게이트웨이에 분배한다.

Step 9 : 홈 게이트웨이는 작성된 사용자 할당 정책을 암호화하여 권한부여 서버에 저장한다.

일반적인 댁내 사용자들에 대해서 사용자들은 자신의 역할을 결정하게 되고, 특수한 상황인 외부 사용자의 경우(예를 들어 가스 검침, 수돗물 점검 등)에는 홈 게이트웨이 오퍼레이터가 그 가이드라인을 제공해줄 수 있도록 한다. 이때 댁내 사용자가 추가되는 경우는 사용자가 알맞은 역할을 할당하면 되고, 삭제되는 경우는 홈 게이트웨이에 있는 정보를 삭제한 다음 다시 암호화하여 오퍼레이터가 보관하고 있는 정보를 업데이트 한다. 외부 사용자의 경우는 오퍼레이터가 사용자 정보의 가이드라인을 주게 되므로, 일단 오퍼레이터의 정보가 수정된 다음, 각각의 홈 게이트웨이에 적용이 된다.

덧붙여, Step 4에서 충돌 검사 시 RBAC에서 권한할

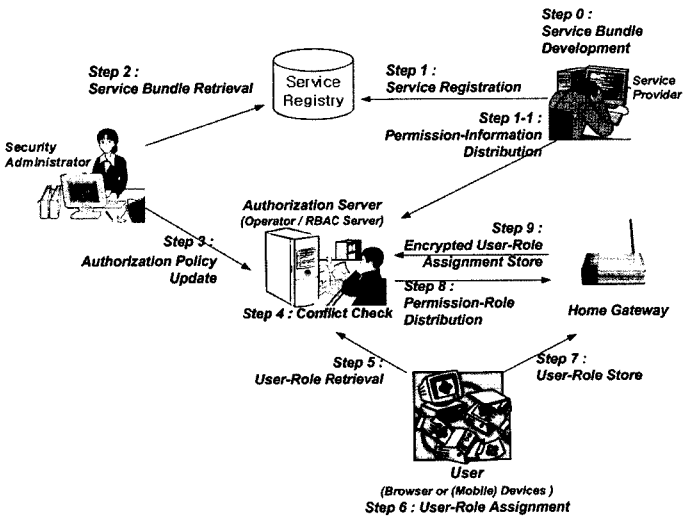

그림 10 제안한 프레임워크에서 권한부여 정보의 운영 과정

당 제약사항 간 충돌 탐지 모델[3]을 사용하여 보안 관리자의 실수를 줄이고 부적절한 권한 할당을 방지하도록 한다. 이 모델은 충돌 탐지 모델을 역할-권한 연관 모델과 권한-권한 연관모델로 나누며, 관계가 포함 관계인지 배제 관계인지, 혹은 연결 관계인지 단절 관계인지 여부에 따라 충돌을 탐지한다.

5.2 사용자 시나리오

댁내 사용자가 서비스를 이용하기 위해서 정책을 활용하는 과정을 순서대로 나타낸 흐름도는 다음의 그림 11과 같다.

사용자가 원하는 서비스를 요청하기 위해서는 먼저 mobile device를 통한 로그인으로 홈 게이트웨이에 접근한다. 홈 게이트웨이는 사용자에 대한 로그인을 통하여 인증 서버에 접속하여 사용자 인증의 과정을 거친다. 사용자가 인증에 성공하지 못하면 재 로그인을 요청하고, 인증에 성공하면 홈 게이트웨이는 그 사용자에 해당하는 사용자 역할 정보를 확인한다. 사용자의 역할이 정의되어 있으면 이용하고자 하는 서비스를 입력하고, 역할이 정의되어 있지 않으면 홈 게이트웨이의 사용자 역할 정보를 수정할 것인지를 확인한다. 새로운 사용자가 추가되는 경우 이 부분에서 사용자 역할 정보를 할당해 주면 된다. 사용자 할당 정책을 수정할 때에는 먼저 권한이 있는지를 확인한 후에 수정을 할 수 있다. 사용자가 입력한 서비스와 사용자의 정보에 정의되어 있는 정책이 일치하면 사용자는 서비스를 제공받을 수 있다. 만약 사용자의 사용자 역할 정보가 정의되어 있지만 사용자가 원하는 서비스가 정의되어 있지 않다면, 그 서비스가 정책에 정의되어 있는지를 확인한 후에 오퍼레이터에게 정책 업데이트를 요청할 수도 있다. 만약 사용자에게 업데이트의 요청의 권한이 없어 오퍼레이트로부터 요청을 거절당한다면, 서비스 제공 역시 거절되고 이 프로세스는 종료가 된다.

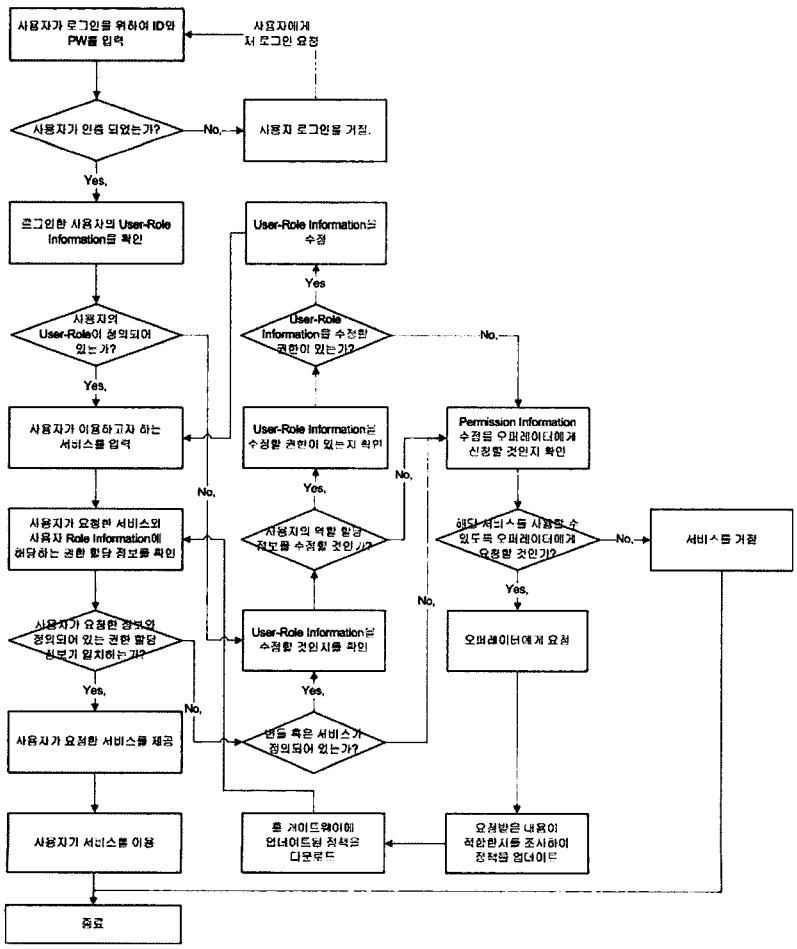

그림 11 사용자 정책 결정의 흐름도

그림 11의 흐름도를 이용하여 접근 제어를 위한 시나리오를 실제 댁내 사용자의 예를 다음에서 서술하였다. 아빠이며 남편인 동시에 관리자 역할을 하는 Robert가 집에서 video service를 주문하여 보기를 기대한다고 가정하고 접근 제어를 위한 가상 시나리오를 제시하였다. 먼저 이 집의 모든 사용자는 사전에 RBAC 서버에 자신의 사용자 할당 정책을 정의하여 저장해 놓았고, 각각의 홈 게이트웨이는 그 가정에 해당하는 사용자에 대한 XML 형태의 권한부여 정보를 RBAC 서버로부터 복사하여 저장하고 있다. 이 경우에서, Robert는 'adult'의 role에 자신을 정의해 놓았고 adult의 role을 가지는 사용자는 모든 종류의 video service를 이용할 수 있다고 설정하고 있다. 또한, 보안 관리자는 인증 서버와 RBAC 서버의 권한부여 정보를 관리한다고 가정한다.

(1) Robert가 video order에 대한 서비스를 요청하기 위해서 자신의 mobile device를 통해 집의 홈 게이트웨이에 접근한다.
(2) 홈 게이트웨이는 Robert에 대한 로그인을 통하여 인증 서버에 접속하여 사용자 인증의 과정을 거친다.
(3) 인증이 성공하면, 홈 게이트웨이는 Robert에 대해서 권한부여 정책을 검사한다.
(4) 홈 게이트웨이로부터 인증 성공이 확인되면, Robert는 video 서비스에 order 서비스를 요청한다.
(5) 홈 게이트웨이와 통신하여 미리 다운 받아 저장해 놓은 권한부여 정보를 통해, 주문된 video가 19세 이상 관람가일 경우 Robert의 요청에 대한 권한 여부, 즉 role name인 adult와 이에 해당하는 bundle service가 적당한지를 확인 한다. 권한과 서비스의 범위가 일치함을 확인되면 요청된 서비스를 제공한다.
(6) 만약 이미 홈 게이트웨이에 저장되어 있던 권한부여 정보와 Robert의 요청을 비교함으로써 Robert의 주문이 적절함을 확인하였다면, 서비스 번들과 정보 가전을 통해서 Robert에게 주문된 서비스를 제공하고, 그렇지 않으면 서비스 제공을 거부한다.

그림 11과 시나리오에서 언급한 바와 같이 실제로 정책의 관리는 서비스를 요청하거나 서비스를 이용하고 있는 도중을 포함한 모든 시간에 항상 일어나는 것이 아니다. 번들이나 사용자에 변화가 있을 경우에만 정책 변경이 일어나게 되는 것이다. 따라서 정책 관리에 대해서는 변경이 일어날 때 오퍼레이터가 일방적으로 배치하기 때문에 서비스를 제공받는 데에는 지연이 일어나지 않지만, 사용자가 서비스를 제공받기 위해 사용자나 서비스의 등록 여부를 확인할 경우에는 지연이 일어나게 된다. 만약 사용자나 서비스의 등록을 하지 않고 마친다면, 지연이 일어나지 않는 대신에 사용자는 등록이 되지 않은 서비스는 제공받지 못하게 되고, 반대로 등록을 하게 되면 등록 시간이 걸리는 대신에 사용자는 원하는 서비스를 제공받을 수 있게 된다.

6. 비교 및 평가

본 절은 ACL과 RBAC 모델 그리고 본 논문에서 제안한 프레임워크에 대하여 정성적인 비교와 정량적인 비교를 한다. 그리고 user-pull, server-pull 구조와 제안한 프레임워크의 변형된 운영 구조를 비교한다. 덧붙여 기존의 접근제어 관련 방법들과 제안된 방법의 비교를 한 뒤, 정적인 상황과 동적인 상황에서의 사용자 요청에 따른 정책 변화를 살펴본다.

먼저, 표 1은 ACL과 RBAC 모델에 대해서 정책의 확장성이 있는지, 정책을 변경할 때 과정과 횟수 면에서 사용자나 개발자들이 편리한지, 정책이 신속하게 업데이트 되는지, 정책을 작성하기에 쉬운지, 정책을 정확하게 작성할 수 있는지, 정책 관리에 있어 경제성이 있는지 등의 측면에서 특성을 비교하였다[31].

표 1 접근 제어 모델의 비교

특징	ACL	RBAC
정책의 확장성	Low	High
정책 변경의 편의성	Low	High
정책 업데이트의 신속성	Low	High
정책 작성의 용이성	High	Low
정책 작성의 정확성	Low	High
정책 관리의 경제성	Low	High

먼저, 정책이 작성될 때 혹은 서비스나 사용자가 추가될 때에 정책을 확장할 필요가 있다. 엔터프라이즈 환경이나 주택 단지의 세대에서 정책이 확장된다면, ACL은 확장되는 부분 하나하나와 관련된 모든 정책을 수정해 주어야하므로 대규모 변경의 경우 정책 관리 시스템의 용량을 늘이는 등의 작업이 필요하다. 반면에 RBAC은 확장되는 부분만을 역할에 할당해주면 되므로 둘을 비교한다면 RBAC이 확장성이 높다. 예를 들어, 서비스가 추가된다고 가정하면, ACL의 경우 정책 작성자는 모든 사용자에 대한 권한을 할당해야만 하는 반면에 RBAC의 경우에는 사용자는 역할만 할당하면 된다. 만약 오퍼레이터나 RBAC 서버에서 정책이 바뀌거나 업데이트 되는 경우를 가정해 본다면, 이때, ACL은 모든 사용자에 대한 권한을 변경해야 하므로 변경 횟수가 많고 시간을 많이 들여야 하는 반면에 RBAC은 서버에서 역할 할당만 변경하여 다운로드 해주면 된다. 따라서 RBAC의 경우가 사용자 편의성과 신속한 업데이트 측면에서 ACL에 비해 더 높은 평가를 할 수 있다. 한편, 작은 규모라면 사용자는 ACL을 사용하는 것이 RBAC보다 더 쉽고 간단할 수도 있다. 작은 규모에서는 ACL을 사용

하여 사용자가 쉽게 서비스에 대한 권한을 작성하고 할당할 수 있기 때문에 사용자는 전문가의 도움 없이 정책을 작성하고 관리할 수 있다. 그러나 관리자의 입장에서는 앞에서 언급했듯이 각 게이트웨이에 해당하는 수많은 사용자와 서비스 번들을 관리해야 하기 때문에 운영이 쉽지 않다. 또한 ACL은 위와 같은 몇몇의 장점을 가지고 있음에도 불구하고 정책을 작성하거나 처리할 때 오류의 가능성이 많다는 단점을 가지고 있다. ACL은 작성이나 수정의 횟수가 훨씬 많고, 사용자가 적을 때에는 간단해 보이기 때문에 사용자가 전문가의 도움 없이 직접 정책을 작성할 수 있기 때문이다. 그러나 RBAC은 사용자가 모든 정책을 작성하는 것이 아니라 자신의 역할만 정의하기 때문에 오류의 가능성을 감소시킬 수 있다. 뿐만 아니라, 정책 관리의 측면에서 ACL은 모든 사용자에 대해서 각각의 서비스들을 정책에 정의해주어야 해서 경제성이 낮다. 반면에 RBAC은 사용자를 역할에 할당해주기만 하면 되기 때문에 ACL에 비해서 경제성이 높다.

표 2는 RBAC 모델을 적용한 일반적인 프레임워크와 제안된 RBAC 모델을 적용한 프레임워크를 비교한 표이다.

표 2 RBAC 기반의 접근 제어 모델의 비교

특징	일반적인 RBAC 프레임워크	본 논문에서 제안한 RBAC 프레임워크
정책 관리의 효율성	High	High
정책의 안정성	High	High
다른 번들과의 연동성	High	High
프라이버시 보호	Low	High
정책 관리의 편의성	Low	High

표 1의 결과는 RBAC의 개념을 도입한 모델은 사용자, 역할의 관리가 기존의 모델보다는 효율적이라는 것을 볼 수 있다. 또한, 표 2에서는 일반적인 RBAC의 정책은 전문가인 오퍼레이터에 의해서 관리되므로 정책의 안정성이 높다. 하지만, 본 논문의 프레임워크 역시 권한할당정보는 오퍼레이터에 의해서 관리되고, 사용자역할정책도 홈 게이트웨이뿐만 아니라 오퍼레이터의 서버에서도 암호화되어 관리되기 때문에 더욱 더 안정적이다. 정책을 관리할 때, 일반적인 RBAC은 사용자 혹은 오퍼레이터가 직접 역할과 권한을 설정하게 되므로 연동의 개념은 해당되지 않고 프라이버시도 보장되지 않지만, 제안된 프레임워크의 홈 게이트웨이에는 사용자 역할 정보를 설정하고 암호화하는 번들만 배치하면 되므로 다른 번들과는 연동이 쉽고 사용자 결정에 있어 프라이버시도 보호할 수 있다. 제안된 모델이 아니라 일반적인 RBAC은 역할과 역할 계층이 구성되어야만 하고 충돌 검사를 받아야만 하기 때문에 약간 복잡하다. 그러나 제안한 프레임워크의 경우, 사용자 측면에서는 역할 할당만 하면 되기 때문에 그것은 문제가 되지 않는다. 제안한 프레임워크에서는, 전문가인 서비스 제공자가 역할과 권한을 정의하는데 관리자 측면에서는 일정한 규칙에 따른 체계적인 관리가 가능하다. 따라서 편의성 측면에서도 제안된 프레임워크가 더 좋은 성능을 보인다 할 수 있다.

현재 사용되고 있는 ACL과 본 논문에서 제안한 RBAC에 대한 정량적인 비교를 하기에 앞서, 먼저 실제적이고 효율적인 비교를 위하여 몇 가지 가정으로 변수들을 고정한 후 수행하였다.

- 각 세대별로 기본적인 사용자 : 4명,
- 각 세대에서 사용하는 기본적인 정보가전 서비스 번들의 서비스 수 : 10개
- 최대 적용되는 세대의 수 : 100세대

여기에서 서비스에 사용자 권한할당 혹은 사용자에게 새로운 서비스 할당을 변경 1로 본다. 추가되는 사용자는 10개의 서비스에 대한 접근을 모두 허가하는 것으로 하고, 추가된 사용자는 일반적으로 하나의 역할에 속하도록 한다.

각각의 집 안에서 개인이 정책을 관리하는 경우와 오퍼레이터가 전문 관리자를 통하여 관리하는 경우를 나누어 비교한다. 오퍼레이터가 관리하는 RBAC의 경우는 제안한 프레임워크를 적용하여 비교한다. 사용자가 추가되거나 정보가전이 변경될 경우 사용자는 그림 11의 흐름도와 같이 변경을 요청한다고 가정한다.

그림 12, 13은 사용자가 추가될 때, ACL과 RBAC의 정책 변경 수를 비교한 그래프이다.

위의 그림 12는 개인이 정책을 관리하는 경우이고, 그림 13은 오퍼레이터가 전문 관리자를 통하여 관리하는 경우이다. 그림 13에서 사용자가 1명 추가될 때, ACL의 경우는 각 가정에 있는 정보가전의 서비스 번들 10개 모두에 대해서 리스트를 변경해 주어야 하고, RBAC의 경우는 추가된 사용자를 하나의 역할에 할당하기만 하면 된다. 따라서 ACL의 경우는 기존에 있던 서비스의 수가 계수가 되어 그 비율로 증가를 하지만, RBAC은 서비스의 수에 상관없이 추가되는 사용자만 역할에 할당해주므로 사용자가 늘어나면 늘어날수록 ACL과 RBAC 사이의 정책 변경 횟수 차이가 커진다.

더 나아가 그림 13과 같이 오퍼레이터의 관점에서 생각해보면, ACL의 경우에 각 게이트웨이 별로 사용자 한 명씩이 추가될 때 100세대 모두에 대해서 정보가전의 모든 서비스 정책 변경을 수행해야하지만, RBAC의 경우에는 제안된 프레임워크를 적용하면 오퍼레이터가 정책을 변경하여 다운로드 해주면 되므로 추가되는 사

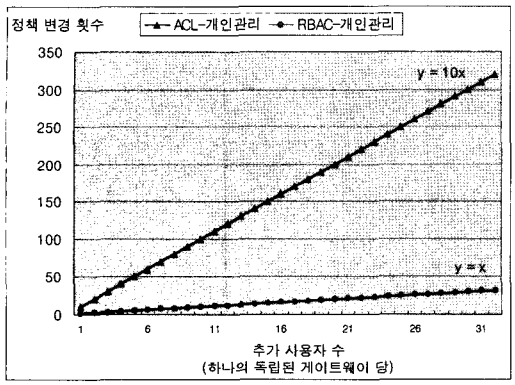

그림 12 개인이 정책을 관리할 때 사용자가 추가되는 경우

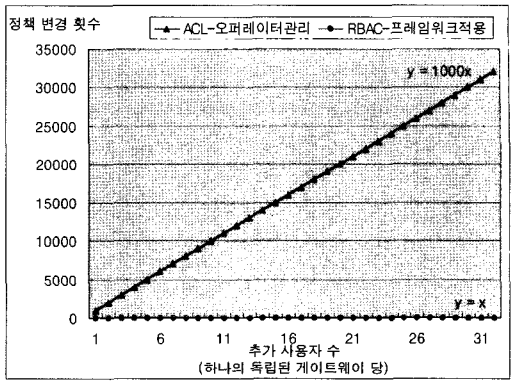

그림 13 오퍼레이터가 정책을 관리할 때 사용자가 추가되는 경우

용자를 역할에 할당해주면 된다. 따라서 ACL의 경우에는 세대의 수와 기존 정보 가전의 수가 모두 계수가 되어 그 비율로 정책 변경 횟수가 증가하고, 제안한 프레임워크를 적용한 RBAC의 경우에는 추가되는 사용자의 수에 따라 변경 횟수가 증가하게 되어 전체적인 정책 변경 횟수의 차이는 더욱 커지게 된다.

위의 그래프 12, 13에 대한 식을 일반화하면 다음과 같은 식을 얻을 수 있다. 정책 변경 횟수를 y, 게이트웨이 i에 추가되는 사용자 수를 a_i, i 홈 게이트웨이에 기존 정보가전의 서비스 수를 b_{iexist}, 적용되는 세대 수를 m이라고 하자.

각 게이트웨이 별로 관리하는 경우는,

$$y_{iACL} = b_{iexist}a_i, \qquad y_{iRBAC} = a_i$$

오퍼레이터가 관리하는 경우는,

$$y_{iACL} = \sum_{i=1}^{m} b_{iexist}a_i, \qquad y_{iRBAC} = a_i$$

로 나타낼 수 있다.

다음 그림 14, 15는 서비스 번들의 서비스가 추가됨에 따라 ACL과 RBAC의 정책 변경 횟수에 대한 비교를 나타낸 그래프이다.

위의 그림 14는 개인이 정책을 관리하는 경우이고, 그림 15는 오퍼레이터가 전문 관리자를 통하여 관리하는 경우이다. 그림 14에서 ACL의 경우, 각 가정에 4명의 사용자 모두에 대해서 서비스 권한을 변경해 주어야 하므로 기존의 사용자 수가 x의 계수가 되어 증가하게 되고, RBAC의 경우 변경되는 서비스를 역할에 할당하기만 하면 되므로 사용자에 관계없이 추가되는 서비스의 수에 따라 변경 횟수가 증가하게 된다. 반면에 그림 15와 같이 오퍼레이터의 관점에서 생각할 때, ACL의 경우에 모든 세대에 대해서 모든 사용자에 대한 정책 변경을 수행해야하므로 사용자와 적용 세대 모두가 정책 변경 횟수와 관련되지만, RBAC의 경우에는 제안된 프레임워크를 적용했을 때, 역시 일괄적으로 변경되는 서

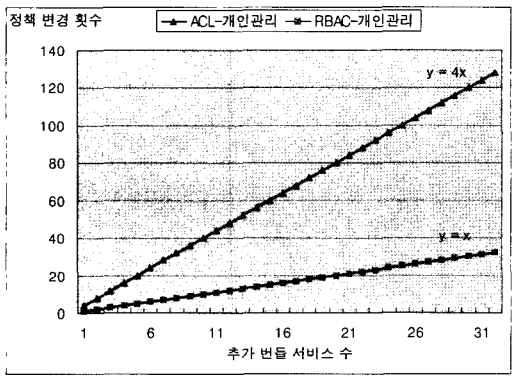

그림 14 개인이 정책을 관리할 때 번들 서비스가 추가되는 경우

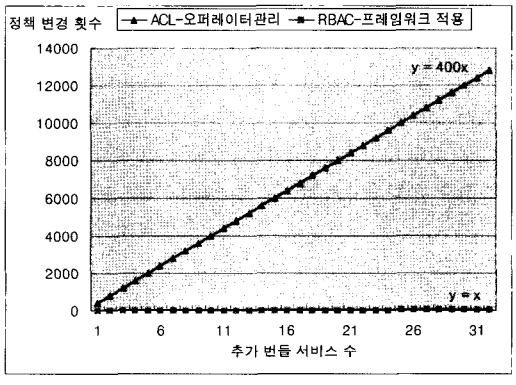

그림 15 오퍼레이터가 정책을 관리할 때 번들 서비스가 추가되는 경우

비스만 역할을 할당해주면 되므로, 추가되는 번들 서비스가 많아지면 많아질수록 그 차이가 커지게 된다.

위의 그래프 14, 15에 대한 식을 일반화하면 다음과 같은 식을 얻을 수 있다. 정책 변경 횟수를 y, 게이트웨이 i에 추가되는 번들 서비스 수를 b_i, 게이트웨이i에 기존의 사용자 수를 a_{iexist}, 적용되는 세대 수를 m이라고 하자.

그러면 각 게이트웨이 별로 관리하는 경우는,

$$y_{iACL} = a_{iexist} b_i, \qquad y_{iRBAC} = b_i$$

오퍼레이터가 관리하는 경우는,

$$y_{iACL} = \sum_{i=1}^{m} a_{iexist} b_i, \qquad y_{iRBAC} = b_i$$

로 나타낼 수 있다.

위의 그림 12~15에서 볼 수 있듯이 두 경우 모두 정책 변경 횟수 면에서 RBAC이 비교적 더 좋은 결과를 나타내주고 있다. 특히 제안한 프레임워크를 적용한 홈 네트워크 환경에서는 RBAC 모델이 훨씬 더 좋은 성능을 보였다.

다음의 표 3은 RBAC 모델에서 user-pull, server-pull 두 가지 운영 구조와 제안된 구조인 변형된 server-pull 구조를 접근에 대한 수행 능력, 역할 및 정책 변경의 신속성, 서비스 이용 시 사용자의 편의성, 프라이버시 그리고 서버의 정책 관리 부담의 면에서 비교하여 설명하고 있다[32].

User-pull 구조의 경우, 접근에 대한 수행 능력은 정책을 관리하는 서버에 요구되는 접근 횟수와 시간 소요의 측면에서 우수하다고 말할 수 있다. 일단 사용자가 인증 정보를 가지게 되면 그 사용자는 정보의 유효기간이 만료될 때까지는 RBAC 서버에 접속할 필요가 없다. 그러나 server-pull 구조의 경우, 애플리케이션은 사용자가 서비스를 요청할 때마다 서버에 접속하는 것이 필요하다. 그래서 시간이 좀 걸리고 서버는 부담을 가지게 된다.

그럼에도 불구하고 일반적으로 말했을 때, 사용자 편의성 측면에서는 server-pull 구조가 user-pull 구조보다 더 낫다. User-pull에서는 인증 정보에 대한 일정한 사용 가능 기간이 있다. 그래서 역할이 변경, 삽입, 혹은 삭제되었을 때, 역할과 정책 변화의 참신성이 감소된다. Server-pull 구조에서는 애플리케이션이 각 세션마다 서버로부터 사용자의 역할 정보를 검색해온다. 이것은 역할 변화에 대한 참신성을 높여서 user-pull 구조보다 정보를 더 효율적으로 업데이트할 수 있다. 이러한 특징들 때문에, server-pull 구조는 특히 동적인 역할 업데이트가 중요한 애플리케이션에 좋은 해결책이 된다[22]. 따라서 '3.2 제안한 프레임워크'는 '(2)권한부여 구조'에서 server-pull 방식을 사용하였는데, 이 방식은 서버에 부하를 많이 주는 취약점을 가지고 있었다. 이를 해결하고자 본 논문에서는 server-pull 구조를 변형하여 적용하였다. 기존에는 권한부여 정보를 RBAC 서버에서 모두 확인해야만 했지만, 변형된 server-pull 구조는 앞의 '3.2 제안한 프레임워크'에서 제시한 바와 같이 권한부여 정보를 RBAC 서버뿐 아니라 홈 게이트웨이에도 위치하도록 한다. 사용자가 서비스를 요청했을 때, 홈 게이트웨이에서 권한 정보를 검사할 수 있도록 한 것이다. 이렇게 하면 서버의 정책 관리 부담을 줄여줄 뿐 아니라 역할 변경의 신속성, 정책 변경의 신속성, 사용자 편의성 등 기존 server-pull 구조의 장점을 모두 가지면서, 접근성, 프라이버시 등을 보장할 수 있다.

다음의 표 4는 접근 제어와 관련된 기존의 연구인 보안 운영체제[5], 자바2 플랫폼[33], XACML 기반 홈 네트워크 접근제어 시스템[34] 등과 제안한 논문을 비교한 것이다.

앞의 표 2와 표 3에서 언급한 바와 같이 제안한 모델은 정책의 확장성, 안정성, 정확성, 경제성, 사용자 편의성, 다른 장치들과의 독립성에 대하여 좋은 평가를 보인다. 이에 반해 보안 운영체제는 주로 하드웨어 리소스에 대한 권한을 관리하기 때문에 응용 서비스에 대한 권한 정책을 설정하기가 어려우며, 정책의 설정이 매우 단순하여 세밀한 접근제어를 하기가 어려우며 프라이버시에 대해서는 고려하지 않고 있다. 자바2 플랫폼의 경우에는 자바의 특성상 하드웨어 및 운영체제와는 독립성을 보장하지만, 자바로만 사용이 가능하므로 사용자가 직접 정책을 정의하는 데에는 어려움이 따르고 다른 시스템에서 사용하는 보안 정책과는 호환 및 연동에 문제가 있다[34]. 또한 XACML 기반의 시스템은 다른 특성에 대해서는 비교적 좋은 성능을 보이지만, 정책 관리에 대

표 3 User-Pull, Server-Pull과 제안된 Server-Pull 구조의 비교

	User-Pull 구조[28]	Server-Pull 구조[28]	변형된 Server-Pull (제안된 구조)
접근에 대한 수행 능력	High	Low	High
역할 변경의 신속성	Low	High	High
정책 변경의 신속성	Low	High	High
서비스 이용 시 사용자 편의성	Low	High	High
프라이버시	Low	Low	High
서버에서 정책 관리의 부담	Low	High	Low

표 4 접근 제어와 관련된 기존 연구와의 비교

비교 항목	제안한 모델	보안 운영체제	자바2 플랫폼	XACML 기반 시스템
정책의 확장성	높음	낮음	낮음	높음
사용자 편의성	높음	낮음	낮음(자바로만 해야됨)	높음
정책의 안정성	높음	높음	높음	높음
하드웨어 및 운영체제와의 독립성	있음	없음	있음	있음
정책 작성의 정확성	높음	낮음	높음	높음
정책 관리의 경제성	높음	N/A	N/A	N/A
사용자 프라이버시 보장	가능	불가능	불가능	불가능

해서 기존의 ACL과 유사한 방법을 사용하고 있어 경제성이 있다고 말할 수는 없으며, 이 모델 역시 프라이버시에 대해서는 고려하지 않고 있다.

다음 그림 16은 정적·동적인 상황에서의 사용자 요청에 대한 정책 변화에 대해서 살펴본 결과이다. 먼저, 정적인 상황은 이미 모든 role을 사용자가 정의하여 서비스를 이용하는데 어떠한 추가적인 요청도 필요하지 않을 때로 정의하고, 동적인 상황은 사용자가 서비스를 요청했으나, 서비스 혹은 사용자에 대한 role이 정의되지 않아 정책의 수정을 요청할 때로 정의한다.

위의 실험은 그림 11의 '사용자 정책 결정 흐름도'를 기반으로 정책 변경 횟수를 카운트 하였다. 먼저, 정적인 상황에서 사용자가 자신의 role에 대한 정책을 변경하기를 원한다면, 자신의 role 정책을 변경할 때, 1번의 정책변경이 일어난다. 반면에 동적인 상황에서, 홈 게이트웨이에 서비스가 정의되어 있지 않을 때, 사용자가 오퍼레이터에게 권한할당정책을 수정해줄 것을 요청한다면, 오퍼레이터는 자신이 가지고 있는 권한할당정책을 수정하고, 수정된 정보를 홈 게이트웨이에 배치하여 사용자할당정책과 매핑한다. 또 다른 경우로, 요청하고자 하는 서비스에 대해서 사용자의 role에 서비스가 정의되어 있지 않을 때, 사용자는 사용자할당정책을 수정한다. 결과적으로 권한 변경과 관련된 정책 변경 요청을 할 때에는 하나의 요청과 관련해 2번의 정책변경이 일어나고, 사용자와 관련된 정책 변경 요청을 할 때에는 1번의 정책 변경이 일어나 정책 변경이 일어날 수 있는 최대

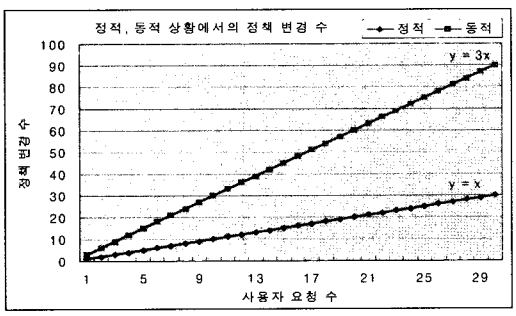

그림 16 정적, 동적 상황에서 정책 변경 수의 비교

의 동적 상황의 경우, 총 3번의 변경이 일어나는 것이다. 따라서 사용자 요청이 많아질수록 동적인 경우 정책 변경의 수가 정적인 경우보다 많아짐을 알 수 있다.

7. 결 론

OSGi 서비스 플랫폼의 사용자 관리 서비스 부분은 프레임워크에 권한부여 정책을 관리하고 운영하기 위해서 자세히 제시되지 않고 있다. 그리고 사용자 접근 제어 부분의 주요 기능 역시 추상적으로 되어있다. 그러나 홈 네트워크의 확산은 이미 본격화 되었고 안전한 사용자 접근에 대한 필요성이 대두되고 있다. 따라서 본 논문에서는 현재의 OSGi 서비스 플랫폼이 운영되고 있는 홈 네트워크 환경에서 명확하지 않은 권한부여 정책과 관리의 부분을 구체화하여 제시하였다. RBAC 기반의 권한부여 정책 관리 방법을 사용하여 OSGi 기반의 홈 네트워크 환경에서 효율적이고 안전한 사용자 접근 제어를 가능하도록 하였다.

본 논문에서는 프레임워크를 제안하기 위해서 필요한 접근제어 모델, 권한부여 구조, 접근제어 정책 작성 및 관리, 접근제어 정책 배치 등의 4가지 고려사항을 제시하여 그 고려사항을 해결하는 접근 제어 프레임워크를 제안하였다. 제안하는 프레임워크의 효율성을 높이고 취약점을 보완하기 위해서 기존의 권한부여 구조를 방법을 변형한 server-pull 구조를 제안하였다. 권한 부여 정책은 효율성과 사용자 편의성 문제를 위해 사용자 할당 정책과 권한 할당 정책으로 나누어 XML 형식으로 작성하고 관리하도록 하였다. 프레임워크의 운영은 제안한 접근제어 프레임워크를 바탕으로 서비스 제공자, 홈 게이트웨이, 사용자, 오퍼레이터, 보안 관리자가 실제 환경에서 효율적으로 구성될 수 있도록 하였으며 더 나아가 비즈니스 모델을 창출할 수 있도록 하였다. 또한 댁내 사용자의 정보는 사용자가 할당하고 권한부여 정보는 오퍼레이터가 관리하므로 프라이버시도 보장하면서 사용자 편의성을 해치지 않고 체계적인 관리를 할 수 있게 하였다.

제안한 프레임워크에서 쓰인 방법들은 기존의 여러 가지 방법들보다 정책 작성의 편의성 측면이나, 효율성

측면에서 높은 결과를 나타내었다. 특히 정책 작성 혹은 수정의 횟수 면에서는 훨씬 경제적인 결과를 확인할 수 있었다.

결과적으로 본 논문은 홈 네트워크에서 권한부여 정책을 체계화 하여 실제 사용 가능한 접근 제어 방법을 제시하였다. 또한 실제 비즈니스적인 면에서도 가이드라인을 제공하였다.

향후 연구로, 인증에 관한 문제 역시 홈 네트워크 보안에서 핵심이 되는 부분이므로 계속해서 연구하고 최적화하는 작업이 필요하다. 또한 이 프레임워크를 시장의 상황에 따라 더욱 구체화하고 상세화할 필요가 있으며, 실제 환경을 기반으로 한 테스트와 다양한 상황에 맞는 정책 형식의 표준화가 필요하다.

참 고 문 헌

[1] OSGi "OSGi Service Platform Release 3 Specification" http://www.osgi.org/. 2006.

[2] 전경석, 문창주, 박대하, 백두권 "OSGi서비스 플랫폼 환경에서의 사용자 인증 메커니즘", 정보과학회논문지, 제9권 제2호, pp. 191-204, 2003.

[3] Chang-Joo Moon, Woojin Paik, Young-Gab Kim, Ju-Hum Kwon, The Conflict Detection between Permission Assignment Constraints in Role-Based Access Control, Lecture Notes in Computer Science, LNCS 3822, pp. 265-278, 2005.

[4] Ant Allan, "Extranet Access Management(EAM): Perspective," Gartner, 2001.

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

[6] 한종욱, 홈네트워크 인증 및 접근제어기술, 홈네트워크 시큐리티 포럼(HNSF), 2004.

[7] 홈네트워크보안연구팀, 홈네트워크를 위한 인증 및 접근권한 제어기술개발, 한국전자통신연구원, 2005.

[8] 한국정보통신기술협회, 홈서버 중심의 홈네트워크 사용자 인증 메커니즘, 정보통신 단체표준 TTAS.KO-12.0030, 2005.

[9] 김재현, 무선 홈 네트워크 환경의 계층별 접근제어, 한국전자통신연구원, 2005.

[10] Dae-Ha Park, Doo-Kwon Baik, OSSEM: a security model for OSGi service framework, 7th World Multi-conference on Systemics, Cybernetics and Informatics (SCI2003), Orlando(USA), pp. 189-194, 2003.

[11] 황지온, 유비쿼터스 환경에 적합한 차세대 홈네트워크를 위한 온톨로지 지식서비스 모델 연구, 중앙대 대학원, 석사학위논문, 2005.

[12] Tao Gu, Hung Keng Pung, Da Qing Zhang, Toward an OSGi-Based Infrastructure for Context-Aware Applications, IEEE Pervasive Computing, Vol.3, No.4, pp. 66-74, 2004.

[13] Harry Chen, Tim Finin, Anupam Joshi, An Ontology for Context-Aware Pervasive Computing Environment, Workshop on Ontologies and Distributed Systems, IJCAI-2003, Acapulco(Mexico), 2003.

[14] Tao Gu, Xiao Hang Wang, Hung Keng Pung, Da Qing Zhang, An Ontology-based Context Model in Intelligent Environments, Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2004), pp. 270-275, 2004.

[15] 박세현, 유비쿼터스 홈을 위한 상황인지 서비스 기술, TTA 저널, 2005.

[16] RDF Resource Description Framework, http://www.w3.org/RDF/

[17] OWL Web Ontology Language, http://www.w3.org/TR./owl-ref

[18] Ravi S. Sandhu, Edward J. Coynek, Hal L. Feinsteink, Charles E. Youmank, Role-Based Access Control Models, IEEE Computer, Vol.29, No.2, pp. 38-47, 1996.

[19] Ravi S. Sandhu, David F. Ferraiolo, Richard Kuhn, The NIST Model for Role-Base Access Control: Toward A Unified Standard, 5th ACM Workshop on Role Based Access Control, Berlin (Germany), pp. 47-63, 2000.

[20] Chang-Joo Moon, Dae-Ha Park, Seong-Jin Park, Doo-Kwon Baik, Symmetric RBAC Model that Takes the Separation of Duty and Role Hierarchies into Consideration, Computers & Security, Vol.23, pp. 126-136, 2004.

[21] Eun-Ae Cho, Chang-Joo Moon, Dae-Ha Park, Doo-Kwon Baik, An Effective Policy Management Framework Using RBAC model for Service Platform based on Components, 4th International Conference on Software Engineering Research, Management and Applications (SERA2006), Seattle (USA), pp. 281-287, 2006.

[22] 김영갑, 문창주, 박대하, 백두권, "OSGi 서비스 플랫폼 환경에서의 서비스 번들 인증 메커니즘의 검증 및 구현", 정보과학회논문지, 제31권 제1호, pp. 27-40, 2004.

[23] David F. Ferraiolo, Role-Based Access Control, Artech House, Computer Security, 2003.

[24] Joon S. Park, Ravi Sandhu, Gail-Joon Ahn. Role-based access control on the Web. ACM Transactions on Information and System Security (TISSEC), Vol.4, No.1, pp. 37-71, 2001.

[25] Sylvia Osborn, Ravi Sandhu, Qamar Munawer, Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies, ACM Transactions on Information and System Security, Vol.3, No.2, pp. 85-106, 2000.

[26] Joon S. Park and Ravi S. Sandhu, Smart certificates: Extending X.509 for secure attribute services on the Web, 22nd National Information Systems Security Conference (NISSC), Crystal City(Virginia), pp. 337-348, 1999.

[27] Joon S. Park and Ravi S. Sandhu. RBAC on the Web by smart certificates, 4th ACM Workshop on

Role-Based Access Control (RBAC), pp. 1-9, 1999.
[28] Joon S. Park and Ravi S. Sandhu, Gail-Joon Ahn, Role-Based Access Control on the Web, ACM Transactions on Information and System Security, Vol.4, No.1, pp. 37-71, 2001.
[29] SAX(Simple API for XML) 2.0.1, "http://www.saxproject.org/," 2006.
[30] DOM(Document Object Model), "http://www.w3.org/DOM/," 2006.
[31] John Barkley, Comparing simple role based access control models and access control lists, 2nd ACM workshop on Role-based access control, Fairfax (USA), pp. 127-132, 1997.
[32] Eun-Ae Cho, Chang-Joo Moon, Dae-Ha Park, Doo-Kwon Baik, Access Control Policy Management Framework based on RBAC in OSGi Service Platform, 6th IEEE International Conference on Computer and Information Technology (CIT06), Seoul(Korea), 2006.
[33] Anne Anderson, Java Access Control Mechanisms, Technical report, Sun Microsystems, "http://lists.oasis-open.org/archives/xacml/200201/pdf00000.pdf," 2002.
[34] 이준호, 임경식, 원유재, XACML 기반 홈 네트워크 접근제어 시스템의 설계 및 구현, 한국정보처리학회 논문지 C Vol.13-C, No.05, pp. 0549-0558, 2006.

조 은 애

2003년 고려대학교 컴퓨터학과 학사. 2005년 고려대학교 컴퓨터학과 석사. 2005년~현재 고려대학교 컴퓨터학과 박사과정. 관심분야는 SSL, 접근제어, 권한부여, RBAC, 홈 네트워크, 프라이버시, 유비쿼터스 보안

문 창 주

1997년 고려대학교 컴퓨터학과 학사. 1999년 고려대학교 컴퓨터학과 석사. 2004년 고려대학교 컴퓨터학과 박사. 2005년 고려대학교 정보보호대학원 연구교수. 2005년 건국대학교 컴퓨터응용과학부 컴퓨터시스템전공 조교수. 2006년~현재 건국대학교 공과대학 항공우주정보시스템공학과 조교수. 관심분야는 접근제어, 권한부여, RBAC, 프라이버시, 유비쿼터스 보안, 임베디드 시스템

백 두 권

1974년 고려대학교 수학과(학사). 1977년 고려대학교 대학원 산업공학과(석사). 1983년 Wayne State Univ. 전산학과(석사) 1985년 Wayne State Univ. 전산학과(박사). 1986년~현재 고려대학교 컴퓨터학과 (교수). 1989년~현재 (사)한국정보과학회 (이사/평의원/부회장). 1991년~현재 (사)한국시뮬레이션학회 (이사/부회장/감사/회장/고문). 1991년~현재 ISO/IEC JTC1/SC32 전문위원회 (위원장). 2002년~2004년 고려대학교 정보통신대학 (초대학장). 2004년~2005년 (사)한국정보처리학회(부회장) 관심분야는 메타데이타, 소프트웨어공학, 데이타공학, 컴포넌트기반 시스템, 메타데이타 레지스트리, 프로젝트 매니지먼트 등