RBAC을 기반으로 하는 향상된 권한 위임 모델

김 태 식[†] · 장 태 무[††]

요 약

역할 기반 접근 제어(Role-Based Access Control)은 역할 계층 구조에서 역할 상속과 의무 분리 등을 제공하여 접근제어의 관리를 쉽게 하는 장점이 있다. 그러나 RBAC은 실 세계에서 빈번하게 이루어지는 권한의 위임을 효율적으로 처리하지 못 한다. 본 논문에서는 위임된 권한의 영속성을 보장하고 최소 권한의 보안 원칙과 의무분리 원칙에 위배되지 않는 향상된 권한 위임 모델(APBDM)을 제안한다. APBDM은 RBAC96을 바탕으로 하며, 사용자 대 사용자, 역할 대 역할의 위임을 제공한다. 위임자는 원하는 권한을 특정인에게 부여 할 수 있고, 위임자가 원하는 시점에서 권한이 회수 될 수 있다. 본 논문에서는 APBDM을 분석하고 이의 유효성을 입증하였다.

키워드 : 역할 기반 접근제어, 역할, 위임, 권한

An Advanced Permission-Based Delegation Model in RBAC

Tae-Shik Kim[†] · Tae-Mu Chang[††]

ABSTRACT

RBAC(Role-Based Access Control) has advantages in managing access controls, because it offers the role inheritance and separation of duty in role hierarchy structures. However, RBAC does not process delegation of permission effectively that occurs frequently in the real world. This paper proposes an Advanced Permission-Based Delegation Model(APBDM) that guarantees permanency of delegated permissions and does not violate security principle of least privilege and separation of duty. APBDM, based on the well-known RBAC96, supports both user-to-user and role-to-role delegation. A delegator can give permission to a specific person, that is delegatee, and the permission can be withdrawn whenever the delegator wants. Our model is analyzed and shown to be effective in the present paper.

Key Words : RBAC(Role-Based Access Control), Role, Delegation, Permission

1. 서 론

인터넷 정보 공유가 급속도로 증가됨으로 인하여 이에 따르는 보안 문제의 해결 방안으로 접근제어(access control) 방식의 필요성이 대두되고 있다. 접근제어 방식들은 임의적 접근제어(DAC: Discretionary Access Control)와 강제적 접근제어(MAC: Mandatory Access Control) 두 가지로 나눌 수 있다. 강제적 접근제어는 관리자가 누가 어떤 정보에 접근 할 수 있는가를 결정하며, 사용자는 그 정책을 변경하는 것이 불가능하다. 임의적 접근제어는 어느 정도의 접근제어를 사용자 또는 객체 접근을 책임지는 관리자에게 그 재량을 두고 사용자는 누가 어떤 객체 접근 권한을 가져야 하는가와 그 권한이 무엇이어야 하는가를 결정할 수 있다. 대부분의 시스템에서 접근제어의 관리를 쉽게 하기 위해 사용자들을 집단으로 묶게 된다. 이 집단은 사용자들의 집단을 대표하게 되며 권한의 집합을 대표하는 것은 아니다. 사용자 집단과 권한 집합 모두 대표할 수 있는 개념으로 역할(role)을 정의하고, 역할을 근거로 접근제어를 수행하는 모델이 역할기반 접근제어(RBAC: Role-Based Access Control)이다[1].

RBAC은 사용자의 역할에 기반이 된 접근통제 방식으로 임의적 접근통제와 강제적 접근통제에 비해 정교함과 유연성을 제공한다. 이의 기본 개념은 권한이 역할과 관련되고 사용자가 역할의 구성을 구축하여 관련된 권한을 추가하는 것이다. 이러한 개념으로 역할 권한 관리를 매우 단순화할 수 있으며, 역할이 조직 내에서 다양한 작업의 기능에 따라 생성되고 사용자의 책임과 자격을 근거로 사용자에게 역할이 할당된다. NIST(National Institute of Standards and Technology)의 RBAC 표준은 사용자 수준의 역할위임에 관하여 정의하지 않고 관리적인 측면에서의 권한 위임만을 정의하고 있다[2, 3]. 위임은 일반적으로 단순위임과 다단계 위임의 두 가지 방법이 존재한다. 단순위임은 자신이 위임 받은 역할을 제3자에게 위임할 수 없다는 것을 의미하고, 다단계 위임은

※ 이 연구는 2004-5학년도 동국대학교 연구년 지원에 의하여 이루어졌음
† 준 회 원 : 동국대학교 대학원 컴퓨터공학과(박사과정 중)
†† 정 회 원 : 동국대학교 컴퓨터공학과 교수
논문접수 : 2006년 4월 3일, 심사완료 : 2006년 8월 11일

위임자의 허가 하에 다시 제3자에게 역할을 위임할 수 있다는 것을 의미한다[6]. 이들 위임 모델 중 대표적인 것으로 RBDM0(Role-Based Delegation Model), PBDM0(Permission-Based Delegation Model), ABDM(Attribute-Based Delegation Model)등이 있다. 그러나 어떤 방법을 사용하더라도 RBAC의 특성상 단순히 역할만을 위임할 경우 피 위임자에게 너무 많은 권한이 위임되게 된다. 또한 사용자는 단지 역할과 관계를 가지며 권한과는 직접적인 관계를 유지하지 않기 때문에 위임 받은 권한에 대한 관리 측면에서도 효율적이지 않다. 더구나 RBAC모델들의 역할 계층에서의 상속 개념과 실질적인 기업간의 조직 관리 규칙이 조화되지 않고 역할 계층에서 볼 때 상위역할과 하위 역할은 서로 의무분리의 관계에 있을 수 없으며 단지 상속 관계일 뿐이다. 그리고 권한 분배의 입장에서 볼 때 권한 위임 시 역할 상속에 의한 연속적인 권한 위임을 유지해야 하는 문제가 발생된다. 또한 PBDM은 위임 시 피 위임자에게 너무 많은 권한이 위임 될 수 있고, 사용자 대 사용자의 위임만을 제공하나 실 세계에서는 역할 대 역할의 위임이 대부분이다.

본 논문에서는 실 세계에서 발생하는 출장이나 휴가 등으로 인한 업무 부재 시 직접 사용자가 위임 할 수 있고, 위임 시 제3자에게 발생되는 과도한 권한 위임을 방지 할 수 있으며 효율적인 관리가 가능한 향상된 권한 위임 모델을 제안하고 이를 평가하고자 한다.

본 논문의 구성은 제안 모델의 연구 배경과 필요성을 1장 서론에서 제시한다. 2장에서는 NIST의 기존 RBAC 모델 및 위임 모델들과 그 문제점에 대해 살펴보고, 3장에서는 본 논문에서 향상된 권한 기반의 위임 모델을 제시한다. 4장에서는 기존의 모델과 본 논문에서 제안하는 모델을 비교 분석하고, 마지막으로 5장에서는 결론 및 향후 연구 과제를 제시함으로써 끝을 맺는다.

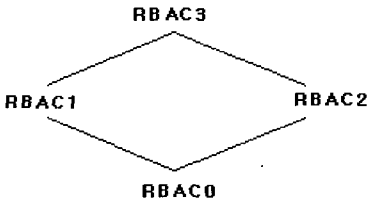
2. 관련 연구 및 연구 동기

본 장에서는 역할 기반 접근 제어의 관련 연구로서 NIST에서 제안하는 표준 참조 모델을 각 단계별로 살펴본다. 또한 기존의 역할 기반의 위임 기법들의 위임할 역할(Delegate-able Role)의 생성과 위임(Delegation), 폐지(Revocation)까지 각 과정별로 알아본다.

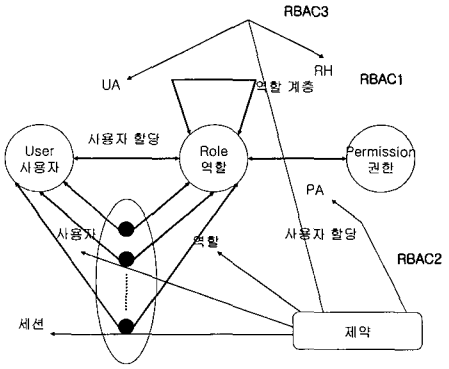
2.1 RBAC

NIST의 표준 참조 모델 RBAC96은 하위부터 RBAC0(Flat RBAC), RBAC1(Hierarchical RBAC), RBAC2 (Constrained RBAC), RBAC3(Symmetric RBAC)로 나뉘고, 하위 단계에서 상위 단계로 올라가면서 하위 단계의 특징들을 내포하게 된다[1].

다음의 (그림 1)은 각 모델간의 관계를 나타낸다. RBAC0의 특성은 각각 상위 모델인 RBAC1과 RBAC2에서 RBAC0의 특성을 내포하게 되고, RBAC3은 그림에서 보는 바와 같이 RBAC1와 RBAC2의 특성을 종합한 모델이 된다. 즉 RBAC3은 하위 모델은 RBAC0, RBAC1, RBAC2의 모든 특성을 갖는 모델이 된다.



(그림 1) RBAC Model간의 관계



(그림 2) RBAC 모델

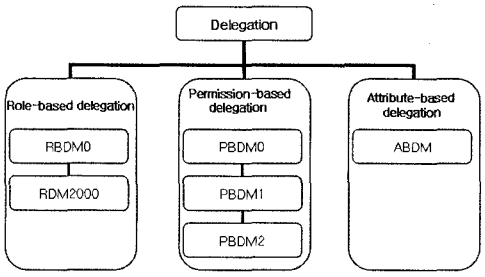
위의 RBAC모델들간의 관계를 바탕으로 RBAC모델은 다음 (그림 2)와 같은 구조를 이룬다. (그림 2)를 바탕으로 각 모델들을 살펴보면, RBAC0에서 요구되는 특성들은 모든 형태의 RBAC에서 필수적인 적으로 다대다(many to many) 사용자 할당 관계와 사용자가 역할을 통해 권한을 얻는다는 점이다. RBAC에는 사용자(User)와 역할(Role), 권한(Permission)이 있고, 사용자는 사람 또는 프로세서가 되며, 역할은 그 멤버에게 수여된 조직 책임과 권한에 관한 의미를 가진 조직내의 직무 기능이나 직무 이름이다. 권한은 권한의 소유자에게 시스템에서 특정한 행동을 수행할 수 있는 능력을 부여한다. 즉 RBAC0는 사용자-역할 할당(UA)과 권한-역할 할당(PA)이 다대다 관계일 것을 요구한다. 또한 RBAC0는 특정한 사용자가 어떠한 역할들에 속하게 되고 특정 역할이 어떠한 사용자들에게 할당 되어지는 가를 효율적으로 결정될 수 있도록 하는 사용자-역할 간 요구사항을 가진다.

RBAC1은 RBAC0에 역할 계층 관계(RH)가 추가된 개념이다. 역할 계층은 조직 내에서 권한과 책임의 순서를 반영하기 위해 역할을 구조화 하는 방법이다.

RBAC2는 위의 (그림 2)에서 보는 바와 같이, RBAC0에 제약조건을 두는 구조로서 제약은 사용자-역할 할당과 사용자 세션 내에서 역할들의 활성화와 관련된다. RBAC3은 RBAC2에 권한-역할 간 요구사항을 추가한 것이다[1-3].

2.2 위임 기법

일반적으로 위임(Delegation)이란 권한의 일부 또는 전부를 제3자에게 주는 것을 의미한다. RBAC에서의 위임은 사용자의 역할이나 권한을 다른 제3자가 가질 수 있게 하는



(그림 3) 위임 기법에 따른 모델 구분

기법이다. 위임이 발생되는 상황은 다음과 같이 세 가지로 나누어 볼 수가 있다.

첫째, 사용자가 오랜 기간동안 자리를 비울 때 업무의 흐름에 지장이 없도록 다른 제3자에게 그 역할의 일부를 임시로 대신 수행하게 하는, 백업(backup)을 생성하는 경우이다. 둘째, 조직을 구성하거나 작업의 효율을 위해서 한 사람 또는 부서에 할당된 권한을 다른 사람에게 재분배하는 경우이다[1-3]. 셋째는 하나의 서비스를 얻기 위해 원격 메소드를 호출하는 경우, 호출한 주체의 권한으로 실행하기 위해 주체의 권한 메소드를 실행하는 사용자에게 부여하는 개념으로서의 위임이며[5][6], 같은 정보를 공유하기 위하여 같은 접근 권한을 갖게 된다. 이와 같은 세 가지 상황에서 역할이나 권한의 위임이 일어나게 된다. 위임 기법 모델들은 역할, 권한, 속성에 따라 (그림 3)와 같이 분류할 수 있다

역할 위임을 기반으로 한 것으로 RBDM0(Role Based Delegation Model)와 RDM2000가 있고, 권한 위임을 기반으로 하는 PBDM(Permission Based Delegation Model), 속성을 기반으로 위임하는 ABDM(Attribute-Based Delegation Model) 등이 RBAC에서의 가장 대표적인 위임 기법으로 볼 수가 있다. RBDM0은 사용자간의 역할 위임을 적용하여 역할을 가지고 있는 사용자가 다른 역할을 가지고 있는 사용자에게 자신의 역할을 위임하는 기법을 제안한 것이고, RDM2000은 RBDM을 확장한 개념으로 역할 계층에서의 정규적인 역할 위임과 다단계 위임을 지원한다. PDBM은 융통성이 있는 위임 모델로서 사용자 대 사용자 간의 위임을 부분적으로 또는 전부를 제공하는 권한을 기반으로 한 권한 위임 기법이다. ABDM은 사용자와 역할 간에 위임을 속성 표현과 전제조건 상태를 제안함으로써 그 속성에 따라 위임하는 기법을 사용한다.

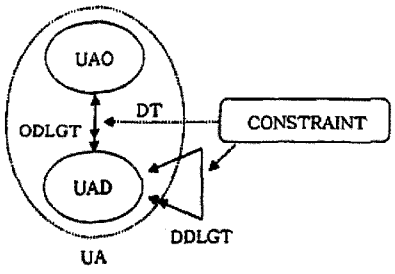
2.2.1 RBAC96에서의 권한 위임

RBAC96모델의 역할계층의 상속 개념과 실제 기업 조직의 관리 규칙은 잘 조화되지 않는다. 즉 상속을 통해 하위 역할이 가진 권한을 상위 역할 자신이 수행 할 수 있도록 권한을 위임하였어도, 역할 상속에 의해 계속 위임한 권한을 유지하게 되는 문제가 생긴다. 이를 방지 하기 위해서는 역할 계층 상에서 상속에 대한 제한을 주어 관리해야 한다. 감독 권한과 같은 제한된 권한에 대해서 상속을 허용하고, 특별한 상황, 예를 들어 역할 담당자의 부재로 인한 역할의 공백을 다루기 위한 백업의 경우 상위 역할로의 한 단계 상속을 허용하여 상위 역할 관리자가 해당 역할을 수행 할 수 있도록 한다. 그러나 현실 세계에서는 상위 역할 담당자가 하위 역할에 대한 작업을 수행하는 일은 거의 일어나지 않는다. 또한 특정 분야의 경우 하위 역할 사용자가 상위 역할 사용자보다 더 전문적 능력을 갖는다. 따라서 단순히 상위 역할이 하위 역할의 백업 역할이 된다는 것은 문제가 있다.

2.2.2 RBDM0에서 권한 위임

RBAC96 계열의 RBAC0에 기반하며 RBDM 모델의 가장 간단한 형태로 위임되고 역할 상속이 이루어지지 않은 형태로 사용자 사이에서 이루어진다. 그 가정과 기본 요소를 보면 동일한 역할을 지닌 사용자간의 위임은 허용되지 않고 일 단계 위임만이 가능하다. 이것은 위임된 역할이 더 이상 다른 사용자에게 위임될 수 없다는 것을 의미하며 원래 구성원만이 위임 할 수 있음을 보여준다. 이러한 위임은 전체적 위임, 즉 위임하는 역할에 있는 개별 사용자는 그 역할에 포함되어 있는 권한의 전체를 위임하거나 전혀 위임하지 않거나 할 수 밖에 없다. 여기서 위임되는 각각의 역할은 시스템 관리자에 의해 역할에 원래 배정된 구성원과 위임받는 구성원으로 나뉜다. 또한 이 모델에서 위임과 철회에 관련된 유일한 요소가 사용자이기 때문에 위임이나 철회에 어떠한 영향도 끼치지 않게 권한을 추가하였다. 위임이 가능한지 또는 불가능한지의 권한을 정의함으로써 관리를 통제할 수밖에 없다. 이런 RBDM을 기반으로 확장된 RDM200은 다음 (그림 4)과 같은 Depth로 위임경로의 깊이를 나타낸다.

RDM2000의 사용자 대 사용자 위임에 대한 위임 사용자, 위임 역할, 위임된 사용자, 위임된 역할 등의 구성요소 사이의 관계를 예를 들면 (그림 4)와 같다. 즉 (Gail, PL2, Dongwa, Q2)의 의미는 Gail이 역할 PL2를 활성화 하여 Dongwa에게 역할 QE2를 위임한다. 위임 관계는 원래 사용자 위임(ODLGT)과 위임된 사용자 위임(DDLGT)으로 분류된다. 이 관계에 기초를 두고 함수를 정의한다. 예를 들면 함수 Prior가 UA(u1, r1)을 다른 UA(u2, r2) 혹은 ø에 사상하기 위해, 또 함수 Path는 위임 경로를 UA을 사상한다. 함수 Depth는 위임경로의 깊이를 반환한다. 위임경로는 순서화된 사용자 역할 배정 관계의 집합이며, 다단계 위임이 적용될 때 위임경로가 생성된다. 위임 경로는 항상 원래 사용자 역할 배정에서 시작한다. 동일한 원래 사용자 역할 배정 위임



(그림 4) 위임 관계

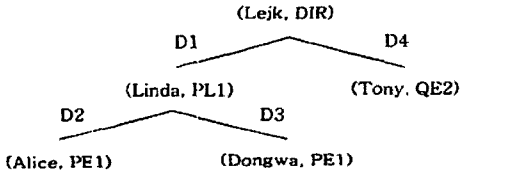
경로는 상위의 위임트리를 구축할 수 있다. 위임 트리는 사용자 역할 배정 및 위임 계층 구조이다. 트리에 있는 사용자 역할 배정의 층은 그 위임 깊이로 지칭된다. 위임 관계는 다음의 집합을 가지고 있다.

D1 : (Lejk,DIR,Linda,PL1) ∈DLGT
D2 : (Linda,PL1,Alice,PE1) ∈DLGT
D3 : (Linda,PL1,Dongwa,PE1) ∈DLGT
D4 : (Lejk,DIR,Tony,QE2) ∈DLGT

위의 위임으로부터 Path 함수를 적용하여 위임경로 P1, P2, P3, P4를 얻을 수 있다. 이 경로에서 관계는 (그림 5)에서 보여주고 있다[8]

계층 구조에 제한을 부과하기 위해 위임 트리의 깊이와 넓이의 한계를 결정해야 한다. 위임의 깊이를 통제하는 데는 비 통제 (no Control), 불리언 통제, 정수 통제가 있으며 비 통제는 역할에 제약이 없고 불리언 통제는 위임을 할 것인지를 결정하여 위임의 최대 깊이에 한계가 있고, 정수 통제는 위임의 최대 깊이를 한정하는 정책을 말한다. 정수 통제를 사용하여 최대 깊이를 제한 할 수 있으나, 위임 관계의 넓이를 통제할 수 없다는 단점이 있다.

DLGT	위 임 경 로
D1	P1: (Linda, PL1), (Lejk, DIR)
D2	P2: (Alice, PE1), (Linda, PL1), (Lejk, DIR)
D3	P3: (Dongwa, PE1), (Linda, PL1), (Lejk, DIR)
D4	P4: (Tony, QE2), (Lejk, DIR)



(그림 5) 위임경로와 위임트리

2.2.3 PBDM에서의 권한위임

위임을 위한 여러 가지 조건이나 제한을 역할 자체와 관련된 속성으로 정의함으로써 위임을 구현하는 방법이다. 즉 위임의 역할을 위한 완전한 접근제어 방법을 위해 다음의 고려 사항을 역할의 속성으로 포함한다.

- 권한 부여 자격-역할은 필요한 자격을 갖춘 사람에게 지정되어야 하는 것처럼 위임 받을 능력도 특정 자격의 소유로 제한되어야 한다.
- 권한 부여 조건-위임 받는 사람의 자격과 함께 역할의 소유자가 더 이상 권한에 대한 책임이 없는 경우와 같은 상황에 대한 권한 부여 조건이 있을 수 있다.
- 위임 집합-어떤 조건에서 위임 될 권한의 집합을 정의한다.
- 위임 단계-위임 정도에 따라 재위임이 일어날 때 기존의 위임 정도에 대한 관계를 명확히 정의한다[9].

각각의 역할이 위의 제약을 속성으로 갖고, 위임이 일어날 경우에 제약 조건을 만족하는 경우에 위임이 이루어질 수 있도록 한다. 이러한 속성의 대부분은 일반성의 손실이 없이 정책 수준에서 구현될 수 있을 뿐 아니라 역할 외부의 정책으로 구현하는 것보다 역할이 가진 속성으로 정의하여 구현할 경우, 인트라넷이나 분산 환경과 같은 좀더 넓은 영역에서 접근제어를 구현하는 유연한 방법을 제공할 수 있다. 이러한 제약은 전적으로 편리하고 관리가 용이하게 기술되어야 한다. 그러나 위임과 관련된 속성뿐 아니라 활성화, 범위 등 여러 부분에 대한 속성을 모두 유지하고, 그러한 속성간의 트리거를 통한 관계를 규정해야 하는 부담을 갖는다.

3. APBDM(An Advanced Permission-Based Delegation Model)

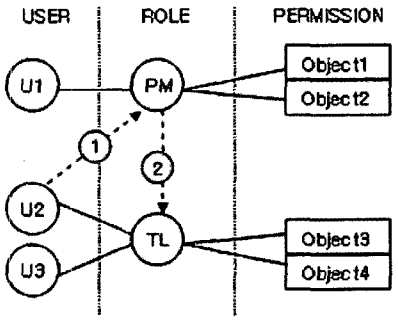
기존의 RBAC모델에서의 권한위임에 관한 연구는 주로 권한 관리자의 업무 분산과 분산된 시스템에서의 권한 위임에 초점을 맞추고 있었다. 하지만 실 세계에서는 사용자가 원하는 위임이 이루어지는 것이 필요하다.

이 장에서는 실 세계의 기업 환경에서의 위임을 분석하고 이를 근거로 운영자 또는 관리자로부터 권한을 부여 받은 사용자가 직접 위임 할 수 있도록 하고, 사용자 부재 시 위임 권한의 영속성을 보장하며 위임의 폐지에서는 위임자와 관리자가 동시에 폐지의 권한을 가질 수 있도록 하는 APBDM을 제안한다.

3.1 실 세계에서 발생되는 위임

위임이란 역할 담당자의 부재로 그 역할을 이행하지 못할 경우에 해당 역할을 제3자가 이행할 수 있도록 자신의 권한의 일부를 임시적 또는 영구적으로 부여하는 것이다. 실제 기업에서는 계획되어 있거나 또는 예기치 않는 일의 발생으로 인한 권한 위임이 빠르게 처리 되어야 하는 상황이 빈번히 발생한다. 이러한 위임이 발생할 때 마다 관리자에게 요청하고 처리되기를 기다리는 것은 업무의 효율성이 떨어질 뿐 아니라 관리자에게는 업무 과부하가 발생된다. 그러므로 사용자는 필요할 때마다 권한 위임의 주체가 되어 언제든지 이행 할 수 있어야 한다.

기존의 RBAC 모델은 관리자와 감사를 용이하게 하는 방법에 초점이 맞추어져 있다. (그림 6)은 기존 RBAC모델에서 위임이 이루어지는 방법을 보인 것이다.



(그림 6) RBAC 구조에서의 위임

첫 번째 방법으로 U2를 U1의 역할 PM에 지정할 수 있다. 이 경우 U2는 역할 PM의 모든 역할을 위임 받게 되어 일부의 권한을 위임하려는 U1의 의도와는 다르게 된다. 두 번째 방법으로는 역할 PM의 권한을 TL에 지정함으로써 실제 TL이 PM의 권한을 가질 수 있도록 할 수 있다. 하지만 이렇게 지정되면 역할 TL을 지정 받고 있는 U3까지 권한을 획득하게 됨으로써 U2에게만 위임하려는 U1의 의도와는 다르다. (그림 5)의 예를 통해 보았듯이, 기존의 RBAC구조는 실 세계에서 일어나는 위임을 제대로 반영하지 못하고 있다. 이렇듯이 실 세계에서 일어나는 위임을 만족하기 위해 최소한 다음과 같은 조건들을 만족해야 한다. 첫 째 운영자 또는 시스템의 간섭 없이 권한 위임이 가능해야 한다[8, 11, 12]. 이는 운영자 또는 시스템의 간섭이 빈번히 이루어진다면 그 만큼의 과부하와 업무상의 차질이 많아 질 수 있기 때문이다. 둘 째 위임자가 원하는 업무에 관한 권한을 특정인에게 부여할 수 있어야 한다[8, 9]. 즉, 현실 세계에서는 상위 역할 담당자가 하위 역할에 대한 작업을 수행하는 일은 거의 일어나지 않고 하위 역할에게만 위임을 부여한다면 이는 위임이라는 개념이라기 보다는 상속의 개념이기 때문이다. 셋 째 위임된 권한은 위임자가 원하는 시점에서 회수될 수 있어야 한다[9, 11]. 넷 째 위임된 권한의 영속성을 보장해야 한다[8], 그럼으로써 위임 권한의 중복성 회피와 사용자 부재 시 위임 권한의 영속성을 보장 할 수 있다. 다섯 째 최소 권한의 보안 원칙에 위배되지 않아야 한다[1, 4, 6]. 여섯 번째 직무분리 원칙에 위배되지 않아야 한다[3, 4, 12].

3.2 APBDM 권한 위임 설계

(그림 7)과 (그림 8)은 APBDM에서의 위임 기법을 설명하기 위한 예이다. (그림 7)에서 사용자 'John'은 'Dir'와 'PM'의 역할을 담당하고 있다. Director에는 'PM', 'BM' 객체에 대한 권한이 할당되어 있고, Staff1에는 'EM'에 접근할 수 있는 권한이 할당되어 있다.

위의 (그림 7)을 보면 다음과 같이 권한 위임을 위한 몇 가지 경우들을 말할 수 있다. Team Leader인 Jenny이 Director인 John에게 단지 DBM에서 권한 W만을 위임하고자 하는 경우와 Jenny이 Tom과 Smith에게 역할DBM에서 권한 W을 Tom에게 위임하고 Smith에게 권한 E를 위임하

사용자(User)

U-ID	Name
U1	John
U2	Tom
U3	Smith
U4	Jenny

역할(Role)

R-ID	Role
R1	Director
R2	Team Leader
R3	Staff1
R4	Staff2

권한(Permission)

R : Read
W : Write
U : Update
D : Delete
E : Execute

객체(Object)

PM	Personnel management
DBM	DB management
De	Development
EM	Enterprise management
BM	Business management

사용자 할당(User Assignment)

U-ID	R-ID
U1	R1
U2	R4
U3	R3
U4	R2

권한 할당(Permission Assignment)

R-ID	Permission
R1	PM [R- - - -]
R2	DBM [RWUDE]
R3	EM [RWU- -]
R4	BM [RWU- -]
R4	De [RW TD -]

(그림 7) 권한 위임의 경우

사용자 할당(User Assignment)

U-ID	R-ID
U1	R1
U2	R4
U3	R3
U4	R2
U1	D1

권한 할당(Permission Assignment)

R-ID	Permission
R1	PM [R- - - -]
R2	DBM [RWUDE]
R3	EM [RWU- -]
R4	BM [RWU- -]
R4	De [RW TD -]
D1	DBM[- - - - E]

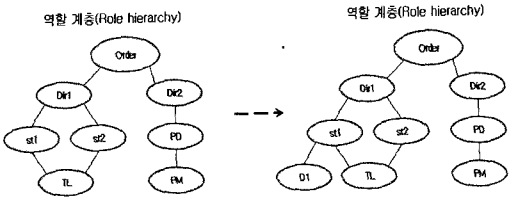
(그림 8) APBDM에서의 위임 예

고자 하는 경우, 그리고, Jenny이 John에게 DBM에서 권한 R과 역할 Staff1을 위임하고자 하는 경우들을 생각할 수 있다. 이중에 마지막 경우를 고려해보자.

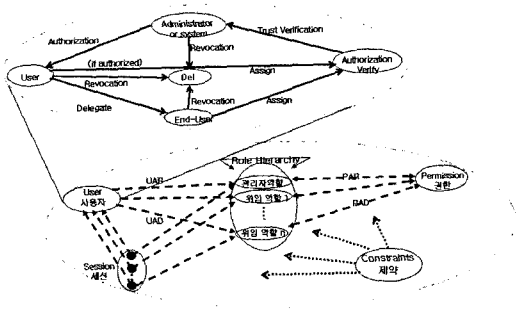
APBDM에서의 역할 생성하고, 그 상위 계층에 제안한 모델을 추가함으로써 허가가 부여된 사용자가 위임할 권한의 집합으로 구성된 새로운 역할을 생성하고 허가가 부여된 위임 권한은 생성한 사용자 및 위임자가 소유권을 가지게 되고, 확인 절차를 거쳐 즉, D1이라는 역할을 새로 생성하여 위임하고 하는 권한을 새로 할당한다. 새로 생성된 역할 D1은 다음의 (그림 8)에서의 관리자 역할에 추가하고 위임하고자 하는 D1역할은 따로 복제하여 두고, 위임자와 관리자가 위임을 주관하여 권한을 할당하여 주게 된다. 새로 생성된 위임 역할을 관리자와 위임자에 의해서 언제든지 폐지가 가능하다. 그에 따른 역할 계층에서의 D1이 추가되는 되는 것은 (그림 9)에서 보는 것과 같다.

(그림 10)는 본 논문에서 제안 하는 위임 기법을 나타내고 있다. 제안하는 기법은 위임하고자 하는 역할에 대해서 역할과 권한을 할당하여 관리자 역할에서 새로 정의를 해주고, 위임하고자 하는 새로운 역할에 대해서는 권한을 할당하고 위임자와 관리자에 의해 위임과 폐지를 관여하게 된다.

다음은 각 모델에 대한 형식적 표현이다.



(그림 9) 역할 계층간의 변화



(그림 10) Advanced Permission-based Delegation Model

3.3 각 모델간의 기본 집합과 함수

U : 사용자 집합 / S : 주체 집합 / R : 역할 / T : 작업 집합
P : 권한 집합 / AR : 관리역할 집합
R' : 역할에 부합된 세부 역할 집합
DRn : 위임역할 집합(n : 0 n : 위임 횟수)
UAR : 사용자에 대한 역할 할당 관계
UAD : 사용자에 대한 위임 할당 관계
PAR : 권한에 대한 역할 할당 관계
PAD : 권한에 대한 위임 할당 관계
PA : 권한에 대한 할당 관계
Authorized-u : 관리자로부터 권한을 부여 받은 사용자의 집합
Assigned-u : 권한을 부여 받은 사용자들로부터 할당된 사용자들의 집합
Assigned-r : 권한을 부여 받은 사용자들로부터 할당된 역할 집합
Assigned-dr : assigned-role로부터 위임된 역할 집합

3.3.1 기본 RBAC96구조

- U, R, P는 각 사용자, 역할, 권한의 집합들과 관계를 갖는다.
- UA ⊆ U X R구조는 사용자와 역할 할당간의 다대다 관계를 갖는다.
- PA ⊆ P X R구조는 권한과 역할 할당간의 다대다 관계를 갖는다.($\forall u \in U$, $\forall r \in R$, UA(r) : UA⊆ U X R)
- UA(r)은 role(r)에 지정된 사용자의 집합을 나타낸다. ($\forall p \in P$, $\forall r \in R$: PA(r)⊆ P X R)
- PA(r)는 role(r)에 지정된 권한의 집합을 나타낸다. ($\forall r1, r2 \in R$: RH(r)⊆ R X R)
- RH는 role(r)과 role(r)사이에 부분적으로 정리된 역할 계층의 집합을 나타낸다.(Users : $R \rightarrow 2^U$)
- 언제든 Users(r)={U|(U,r)∈UA}이라면 사용자의 집합에게 각각의 역할 r에 일치된 UA로부터 기능이 결정된다. (Permission : $R \rightarrow 2^P$)
- 언제든 Permission (p)={P|(P,r)∈PA}이라면 권한의 집합을 각각의 역할 r에 일치된 PA로부터 기능이 결정된다. ($U \rightarrow 2^S$)
- Sessions은 주체의 집합 S를 사용자에게 일치 시키는 것을 나타낸다.($S \rightarrow 2^R$)
- Role은 역할의 집합을 주체에게 일치 시키는 것을 나타낸다.($S \rightarrow 2^P$)
- 권한은 권한의 집합에 각각의 주체를 일치시키는 PA로부터 결정됨을 나타낸다.

3.3.2 RBDM0 모델.

UAO⊆ U X R
-실질적인 구성원과 역할 할당간의 다대다 관계를 갖는다.
UAD⊆ U X R
-위임 구성원과 역할 할당간의 다대다 관계를 갖는다.
UAO ∩UAD =Ø
-같은 역할이 분리 되어지는 실질적인 구성원과 위임 구성원을 나타낸다.
User_O(r)={U|(U,r)∈UAD}
-역할에서 모든 구성원 User_O(r) ∪ Users_D(r)은 역할에 할당된 모든 권한을 갖는다.
-T는 그 역할의 존속된 시간의 집합을 나타낸다.
-위임 역할 즉, UAD → T는 단기간 동안 각각 위임의 일치 기능을 나타낸다.

3.3.3 제안 모델 구조

[정의1] UAR은 assigned-role(r)에 지정된 사용자 authorized-u는 위임 역할을 생성 할 수 있는 권한을 갖는다.
-UAR ⊆ U X R
$\forall$assigned-r1 ∈ R, $\forall$authorized-u∈UA(r) : Create_dr (assigned-r1,assigne-dr2) →r1∈RU(u)
[정의2] UAD은 권한을 부여 받은 사용자 authorized-u는 자신이 가진 역할 r의 일부분에 대해 위임 할 수 있다.
-UAD ⊆ authorized-u X DRn
$\forall$authorized-u ∈ UA(r) : Select_R'((r'), t) → t∈subset of RR'T(r')
[정의3] PAR은 역할 r에 지정된 권한 p을 모두 포함한다.
-PAR ⊆ P X R
[정의4] PAD은 위임 역할 DRn에 지정된 권한 p을 모두 또는 일부 포함한다.
-PAD ⊆ P X DRn
$\forall$assigned-r1, dr2 ∈ R, $\forall$p1, p2⊆ P, $\forall$authorized-u ∈ UA(r), assigned-r1>dr2
p1⊆PAR(dr2)∧p2⊆PAR(dr2)∧p1⊆PDR(dr2)∧p2⊈PDR(dr2)∧assigned-r1∈UAD(u)→(p2⊆PDR(dr2)→dr2∈UDR(u))

이와 같이 [정의4]는 상속 관계를 정의하여 역할의 중복을 방지한다. 또한 위의 모든 정의의 기본 형식은 일반적인 RBAC의 사용자와 역할 그리고 역할과 역할의 관계를 나타낸 정의와 형식에 따른다.

3.4 위임 역할의 생성

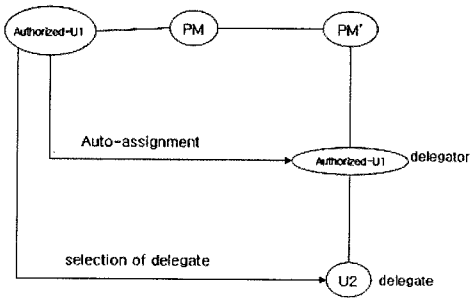
위임의 주체는 사용자이다. 하지만 사용자는 단지 역할에 포함되고 역할에 할당된 권한을 가질 뿐이다. RBAC으로 구현된 기업의 조직에서 이루어지는 위임에 대한 적절한 모델은 다음과 같다.

1) 단순 위임(simple delegation) : 위임자는 임의적 권한을 다른 객체에 부여할 수 있는 유일한 사람이다. 즉 해당 권한을 위임 받는 사람이 제 3자에게 또 다시 권한을 부여 할 수 없다.
2) 다단계 위임(liberal delegation) : 위임 받는 사람은 또 다른 제3자에게 위임 받을 권한의 일부 또는 다시 위임 할 수 있다.
3) 권한위임과 독립된 권한 회수(delegation-independent

revocation) : 권한을 부여한 사람과 관계 없이 회수에 대한 권한을 가진 사람은 누구나 부여된 권한을 회수할 수 있다.

관리자로부터 권한을 부여 받은 권한을 위임하려는 사용자는 위임할 권한을 포함하는 새로운 역할을 생성할 수 있다. 새로운 역할은 사용자가 이미 지정되어 있는 역할의 부분집합이 된다. 이렇게 생성된 역할은 역할을 생성한 사용자의 소유가 된다.

(그림 11)과 비교해서 살펴보면 권한을 부여 받은 PM인 U1은 자신의 역할의 일부분을 위임하려고 한다. 이때 PM인 U1은 자신의 역할의 일부분을 위임하려고 한다. 이때 PM의 부분집합인 PM'이 생성됨과 동시에 U1은 delegete에 지정된 U2는 제 3자에게 다시 위임할 수 있다. 이를 테이블로 정리하면 다음과 같다.



(그림 11) 제안 모델에서의 위임

〈Table 1〉 위임 생성을 위한 테이블

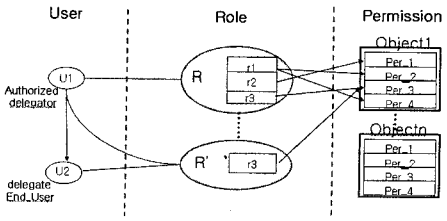
delegation	Delegator	Delegate
r'(PM')	Authorized-u1	U2

새로 생성된 위임역할을 관리하기 위해 자동으로 위임자 역할인 delegator-user와 위임 받는 사용자 역할인 delegate_user 역할을 생성하며 이들 역할에 해당하는 허가도 생성한다.

3.5 지정된 권한만의 위임

실 세계에서 접근 권한을 부여하는 실제적인 단위는 역할이 아닌 업무이다. 따라서 하나의 역할에는 하나 이상의 업무가 포함된다. 하지만 업무 단위로 권한관리를 하는 것이 불가능하다. 또한 실 세계에는 여러 특성을 갖는 업무들이 존재하고, 그 특성에 따라 서로 다른 관리를 필요로 하는데 RBAC은 이를 지원하지 못한다.

그런 이유로 제안된 모델에서의 역할은 업무의 집합을 의미한다. 즉 역할과 관련된 작업의 세분화를 통해서 권한의 일부분을 위임 할 수 있도록 한다. (그림 12)에서 PM의 작업 r1(설계), r2(분석), r3(구현)으로 나눌 수 있고, 관리자로부터 권한을 부여 받은 U1은 U2에게 r3에 관한 권한만을 위임하기를 원한다고 가정한다. 변경된 의미의 역할에서는 PM을 작업의 단위로 세분화했기 때문에 원하는 업무 r3만을 선택해서 위임 할 수 있다.



(그림 12) 변경된 의미의 역할 위임

3.6 위임 감독

이제까지 제안한 모델의 상위 계층에서는 위임될 대상에 대해 권한을 부여 받은 사용자와 기존의 역할이 갖는 제약조건을 상속하여 제한함으로써 시스템 상에서 위임 대상에 대한 제한적 선택을 가능하게 한다. 그러나 기준을 만족하는 대상 사이에서 위임이 일어나더라도, 감독 허가와 같은 경우 조직의 하위 위임을 해서는 안 되는 권한 위임이 일어날 수 있다. 또한 위임 받는 제3자가 그들의 의무를 적절하게 이행하지 않을 위험이 있다. 따라서 사용자의 행위가 다른 제 3자에 의해 감독되어야 하며, 같은 역할 계층에서 일어난 위임도 이러한 위임이 적합한가에 대한 동일 범위 여부에 대한 판단을 해야 한다.

위임 역할은 새로 생성된 역할이기 때문에 다른 역할과의 계층관계를 갖지 않는다. 따라서 그러한 역할을 감독할 만한 역할이 존재 하지 않는다. 이런 점을 보안하기 위해 상위 계층에서 사용자에 대한 검증을 하고 위임 역할을 생성한 역할이 위치한 역할 계층에서 상위 역할이 가지는 감독, 권한에 대한 정보를 상속 받는다. 따라서 새로 생성된 위임 역할에 대한 감독은 기존 역할에 대한 감독을 하고 역할 계층상의 상위 역할이 위임역할에 대한 감독도 한다.

위임 역할에 대해 사용자 할당을 한 경우, 이는 정적 할당이 일어난 것이다. 관리자로부터 권한을 부여 받은 사용자가 위임 역할을 활성화 하여 해당하는 허가를 이행하려면, 사용자 세션 내에서 해당 역할을 활성화하기 위해 상위 감독 역할 사용자에 의해서 정적 역할-사용자 할당이 적합함을 승인 받은 후에서야 가능하다. 상위 감독 역할의 승인 없이 위임 역할에 대한 할당이 일어나면 실제 할당된 사용자가 그 역할을 세션 내에서 활성화 시키지 못하여 작업을 할 수 없다. 즉 위임 역할을 이행 할 수 없다.

4. PBDM 모델과의 비교 분석

다음의 <표 1>은 기존의 위임 기법과의 제안한 기법간의 위임 유형, 위임 역할 회수, 위임 거부, 부분 권한 위임, 위임 관점 등을 비교 분석한 결과이다.

위의 <표 1>에서 보는 바와 같이 PBDM모델에서는 사용자위임 또는 새로운 역할을 생성하여 역할 위임을 지원한다. 반면에 제안한 기법에서는 기존의 RBAC과 PBDM 모델에 위임의 특성을 추가하여 권한의 부분적인 위임이 사용자와 관리자의 권한으로 위임을 가능하게 하였다.

〈표 1〉 기존 모델들과 제안 모델의 위임 비교

비교 기준		RBAC96	RBDM0	PBDM0	APBDM
위임	위임 유형	지원하지 않음	단순위임	단순 위임	단순위임 다단계 위임
	위임 역할 회수	지원하지 않음	관리자	관리자	위임자 관리자
	위임 거부	지원하지 않음	지원하지 않음	지원하지 않음	피 위임자
	부분 권한위임	지원하지 않음	지원하지 않음	권한 부분 집합	권한 부분 집합
	위임 관점	지원하지 않음	사용자	사용자	사용자 관리자

〈표 2〉 기존 RBAC모델과 제안 모델의 무결성 비교

비교기준		RBAC0	RBAC1	RBAC2	APBDM
무결성	임무 분리	지원하지 않음	지원하지 않음	SSD DSD	SSD DSD
	권한 분리	지원되지 않음	지원하지 않음	지원하지 않음	SSP DSP
	사용자수	최대 사용자 수 제한	최대 사용자 수 제한	최대 사용자 수 제한	최대 위임자 수 제한

위의 <표 2>는 기존 RBAC 모델과 제안 모델의 무결성 측면을 비교하였다. 표에서 보는 바와 같이 제안 기법은 정적 의무 분리를 지원하고, 사용자 수에서는 관리자의 역할을 복제하여 위임을 하기에 최대 위임자 수를 제한하였다.

5. 결론 및 향후 연구 과제

본 논문에서는 기존의 RBAC 표준 참조 모델을 기반으로 위임 기법을 추가하여 모델링 하여 보았다. 즉, 권한 위임에 중점을 두어 관리자는 모든 역할과 권한을 관리할 수 있는 역할을 가지며, 각 사용자는 자신의 권한을 제3자에게 위임을 하고자 하는 경우에는 관리자의 역할에서 위임자가 위임하고자 하는 역할을 복제하여 위임을 하게 된다. 또한 위임의 폐지에서는 위임자와 관리자가 동시에 폐지의 권한을 가지게 되는 위임 기법을 제안하였다.

본 논문에서의 위임 기법을 기존의 RBAC 표준 참조 모델과 비교 분석하여 기존의 RBAC을 바탕으로 부분 권한 위임이 추가됨과 임무 분리와 권한 분리가 지원됨에 있어서 그 규칙에 위배되지 않음을 보였다.

향후 연구과제로는 제안한 모델을 바탕으로 권한을 조금 더 세부적으로 제한하여 권한 위임에 있어서 강화된 접근제어 모델을 연구 할 것이며, 비교분석에서는 위임기법이 적용된 다른 모델들과 비교하고자 한다.

참 고 문 헌

[1] D. Ferraiolo And D.R.Kuhn, "Role-based access controls", 15th NIST_NICS National Computer Security Conference, pp.554-563, Baltimore, MD, October 13-16, 1992.

[2] D. Ferraiolo , J. Cugini and D.R.Kuhn, "Role-based Access Control : Features and Motivations", In Annual Computer Security Applications Conference, pp.241-248, November 09, 1995.

[3] Ravi Sandhu, Edward j. Coyne, Hal L. Feinstein and Charles E. Youman, "Role-based Access Control Model", IEEE, pp.38-47, Feb., 1996.

[4] Ezedin Barka and Ravi Shanhu, "A Role-Based Delegation model and Some Extensions", Proc, Of 23rd National Information System Security Conference(NISSC 2000), pp.168-176, December, 2000.

[5] Ezedin Barka and Ravi Shanhu, "Framework for Role-based Delegation Model and Some Extensions", Proceedings of the 23rd NIST-NCSC National Information Systems Security Conference, pp.101-114, Baltimore, USA, October, 2000. 3.

[6] Gail-Joon Ahn, "Specification and Classification of Role-based Authorization Policies", In Proceedings of 8th IEEE International Workshop on Enterprise Security (WETICE2003), pp.202-207, June 9-11, 2000.

[7] Gail-Joon Ahn and Ravi Shanhu, "Role-based Authorization Constraints Specification", ACM Trans on Information and System Security, Vol.3, No.4, pp.207-226, November, 2000.

[8] Zhang L, Gail-Joon Ahn and Chun B.T, "A Rule-based Framework for Role-based Delegation Revocation", ACM Transactions on Information and System Security , Vol.6, No.3, pp.404-441, August, 2003.

[9] XinWen Zhang, Sejong Oh and Ravi Sandhu, "PBDM: A Flexible Delegation Model in RBAC", 8th ACM Symposium on Access Control Models and Technologies(SACMAT -03), pp.149-157, June, 2003.

[10] Chunxiao Ye, Yunqing Fu, Zhingfu Wu, "An attribute-Based-Delegation-Model", ACM International Conference Proceeding Series,Vol85, Proceedings of the 3rd international Conference in Information security, pp.220-221, November 14-16, 2004.

[11] Nighui Li, Mahesh V, Triounitara, "Security Analysis in Role-Based Access Control", Proceeding s of the Ninth ACM Symposium in Access Control Models and Techniques (SACMAT 2004), pp.126-135, June 2-4, 2004.

[12] Serban I, Bavrila, Jogn F, Barklev, "Formal Specification for Role Based Access Control User/Role and Role/Role Relationship Management", ACM Workshop on Role-Based Access Control, pp.81-90, 1998.

김 태 식

e-mail : creta74@dongguk.edu

2003년 동국대학교 컴퓨터멀티미디어 공학과(공학사)

2006년 동국대학교 대학원 컴퓨터공학과 (공학석사)

2006년~현재 동국대학교 대학원 컴퓨터 공학과(박사과정 중)

관심분야 : 정보보안, 컴퓨터 네트워크, Delegation, Permission, Embedded, Ubiquitous 등

장 태 무

e-mail : jtm@dongguk.edu

1977년 서울대 전자공학과(공학학사)

1979년 한국과학기술원 전산학과 (공학석사)

1995년 서울대 대학원 컴퓨터공학과 (공학박사)

1979년~1981년 한국전자기술연구소 연구원

1981년~현재 동국대학교 컴퓨터공학과 교수

관심분야 : 정보보안, Embedded, Ubiquitous, Ad-Hoc, Grid 등