韓國컴퓨터情報學會 論文誌
第15卷 第11號, 2010. 11.

2010-15-11-3-6

RBAC을 기반으로 하는 시간제한 권한 위임 모델

김 태 식*, 장 태 무**

A Time Constraints Permission Based Delegation Model in RBAC

Tae-Shik Kim*, Tae-Mu Chang**

요 약

역할 기반 접근제어는 역할 계층 구조에서 역할 상속과 의무 분리 등을 제공하여 접근제어의 관리를 쉽게 하는 장점이 있다. 위임은 사용자에게 접근 권한을 할당하는 메커니즘이다. RBDM0과 RDM2000은 사용자 대 사용자 위임으로서의 역할 위임이다. 그러나 RBAC은 실세계에서 빈번하게 이루어지는 역할이나 권한위임을 효율적으로 처리하지 못 한다. 본 논문에서는 위임된 권한의 영속성을 보장하고 최소 권한의 보안 원칙과 의무분리 원칙에 위배되지 않는 시간제한 권한 위임 모델(TCPBDM)을 제안한다. TCPBDM은 RBAC96을 바탕으로 하며, 위임에 시간제한과 함께 사용자 대 사용자, 역할 대 역할의 위임을 제공한다. 위임자는 원하는 권한을 특정인에게 부여 할 수 있고, 시간제한을 활용하여 위임자가 원하는 시점에서 권한이 회수 될 수 있다. 본 논문에서는 TCPBDM을 분석하고 이의 유효성을 입증하였다.

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. Delegation is a mechanism of assigning access rights to a user. RBDM0 and RDM2000 models deal with user-to-user delegation. The unit of delegation in them is a role. However, RBAC does not process delegation of Role or Permission effectively that occurs frequently in the real world. This paper proposes a Time Constraints Permission-Based Delegation Model(TCPBDM) that guarantees permanency of delegated permissions and does not violate security principle of least privilege and separation of duty. TCPBDM, based on the well-known RBAC96, supports both user-to-user and role-to-role delegation with time constraints. A delegator can give permission to a specific person, that is delegatee, and the permission can be withdrawn whenever the delegator wants.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.

▸ Keyword : RBAC96, Delegation Model, Time Constraints, TCPBDM

• 제1저자 : 김태식 교신저자 : 장태무
• 투고일 : 2010. 09. 13, 심사일 : 2010. 10. 04, 게재확정일 : 2010. 10. 13.
* 동국대학교 컴퓨터공학과 박사수료 ** 동국대학교 IT학부 컴퓨터공학과 교수

Ⅰ. 서 론

인터넷 정보 공유가 급속도로 증가됨으로 이에 따르는 보안 문제의 해결 방안으로 접근제어(access control) 방식의 필요성이 대두되고 있다. 이런 접근제어 방식들은 임의적 접근제어(DAC: Discretionary Access Control)와 강제적 접근제어(MAC: Mandatory Access Control) 두 가지로 나눌 수 있다. MAC은 관리자가 누가 어떤 정보에 접근 할 수 있는가를 결정하며, 사용자는 그 정책을 변경하는 것이 불가능하다. DAC은 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].

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 단순위임과 다단계 위임의 두 가지 방법이 존재한다. 단순위임은 자신이 위임 받은 역할을 제3자에게 위임할 수 없다는 것을 의미하고, 다단계 위임은 다시 제3자에게 역할을 위임할 수 있다는 것을 의미한다[5][6]. 이들 위임 모델 중 대표적인 것으로 RBDM(Role-Based Delegation Model), PBDM(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 비효율적이다. 그리고 권한 분배의 입장에서 볼 때 권한 위임 시 역할 상속에 의한 연속적인 권한 위임을 유지해야 하는 문제가 발생된다.

따라서 본 논문에서는 실세계에서 발생하는 출장이나 휴가 등으로 인한 업무 부재 시 직접 사용자가 역할을 통해 자신의 권한을 위임 할 수 있고, 위임 시 제3자에게 발생되는 과도한 권한 위임이나 악의적인 권한 위임을 방지 할 수 있으며 효율적인 관리가 가능한 시간제한 권한 위임 모델을 제안하고 이를 평가하고자 한다.

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

Ⅱ. 관련 연구 및 연구동기

본 장에서는 역할 기반 접근 제어의 관련 연구로서 NIST에서 제안하는 표준 참조 모델을 각 단계별로 살펴보고 기존의 위임 기법모델들과 위임에 따른 문제점을 알아본다.

1. RBAC96

NIST의 표준 참조 모델 RBAC96은 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].

그림 1은 RBAC모델들 간의 관계를 바탕으로 RBAC모델을 보인 것이다. RBAC의 구성 요소에는 사용자(User)와 역할(Role), 권한(Permission)이 있고, 사용자는 사람 또는 프로세서가 되며, 역할은 그 구성원에게 수여된 조직책임과 권한에 관한 의미를 가진 조직 내의 직무 기능이나 직무 이름이다. 권한은 권한의 소유자에게 시스템에서 특정한 행동을 수행할 수 있는 능력을 주는 수단이다.

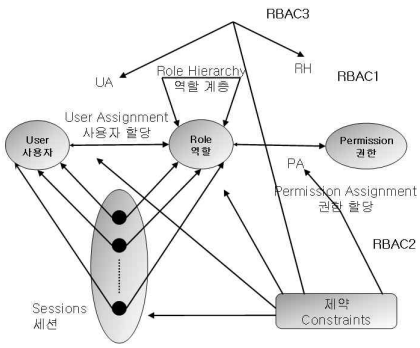

그림 1. RBAC96 모델
fig. 1. RBAC96 Model

RBAC0은 사용자-역할 할당(UA)과 권한-역할 할당(PA)이 다대다 관계를 요구한다. RBAC1은 RBAC0에 역할 계층

(RH) 관계가 추가된 개념이다. 역할 계층은 조직 내에서 권한과 책임의 순서를 반영하기 위해 역할을 구조화한 방법이다. RBAC2는 위의 그림 1에서 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][2][3].

2. 위임 기법

위임이 발생되는 상황은 다음과 같이 나누어 볼 수가 있다.

첫째, 사용자가 오랜 기간 동안 자리를 비울 때 업무의 흐름에 지장이 없도록 다른 이에게 그 역할의 일부를 임시로 대신 수행하게 하는, 백업(backup)을 생성하는 경우이다. 둘째, 조직의 구성 또는 작업의 효율을 위해서 사람 또는 부서에 할당된 권한을 다른 사람에게 재분배하는 경우이다[1][2][3]. 셋째는 하나의 서비스를 얻기 위해 원격 메소드를 호출하는 경우, 호출한 주체의 권한으로 실행하기 위해 메소드를 실행하는 사용자에게 권한을 부여하는 개념으로서의 위임이 된다[6][7]. 이와 같은 상황에서 역할이나 권한의 위임이 일어나게 된다.

위임 기법 모델은 역할, 권한, 속성 위임에 따라 다음과 같이 역할 위임을 기반으로 하는 RBDM0와 RDM2000이 있고, 권한 위임을 기반의 PBDM, 속성을 기반의 ABDM들이 분류된다.

2.1 RBAC96에서의 위임

RBAC96의 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 위임한 권한을 지속적으로 유지하게 되는 문제가 생긴다. 이를 방지하기 위해서는 역할 계층상에서 상속에 대한 제한을 주어 관리해야 한다. 감독 권한과 같은 제한된 권한에 대해 상속을 허용하고, 상위 역할로의 한 단계 상속을 허용하여 상위 역할이 해당 역할을 수행 할 수 있도록 한다. 그러나 현실 세계에서는 상위 역할 담당자가 하위 역할에 대한 작업을 수행하는 일은 거의 일어나지 않는다. 따라서 단순히 상위 역할이 하위 역할의 백업 역할이 된다는 것은 문제가 있다.

2.2 RBDM0에서 위임

RBAC96에서 RBAC0에 기반하며 위임 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 위임할 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. 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 RDM2000은 다음 그림 3과 같은 Depth로 위임경로의 깊이를 나타낸다.

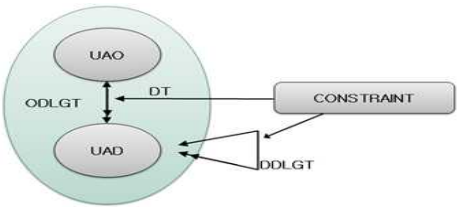

그림 3. RDM2000에서의 위임 관계
fig 3. Delegation relationship in RDM2000

이런 위임 관계는 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.

2.3 PBDM에서의 권한위임

RDM2000을 기반으로 확장하여 사용자-사용자 권한을 위임하는 모델로서 PBDM0, PBDM1, PBDM2로 나뉜다. 사용자가 권한 위임을 위해 임시 역할인 RR(Regular Role)을 생성하고 위임할 권한을 할당하여 위임한다. RR에 권한을 할당하여 위임하기 때문에 위임 해제 시에는 RR 자체를 제거하거나 할당된 권한 중 일부만을 제거하면 위임을 해체할 수 있기 때문에 효율적이다. 하지만, 사용자에 의해 생성된 역할은 관리자가 관리를 하지 못함으로 악의적으로 사용자가 권한을 위임 할 수 있다. 이러한 단점이 있다. PBDM1은 PBDM0를 확장한 모델로써 권한을 위임이 가능한 역할 DBR(Delegable

Role)과 불가능한 역할 RR으로 나눈다. 사용자는 권한 위임을 위해 DTR (Delegation Role)을 생성한 다음, DBR에 할당된 권한 중 위임할 권한을 DTR에 할당하여 위임한다. 위임을 위해 생성된 DTR은 생성한 사용자가 관리하게 된다. PBDM2는 PBDM0을 확장함으로써 역할-역할 위임으로 PBDM1과 같이 역할의 권한을 위임 가능한 권한과 위임 불가능한 권한으로 나뉜다. RR에 할당된 권한은 위임이 불가능하고, FDBR(Fixed Delegable Role)에 할당된 권한은 위임이 가능하다.

TDBR(Temporal delegable role)은 임시 역할 위임으로서 역할-역할 할당을 가진 위임자로부터 권한을 위임 받거나, FBDR에 의해 권한을 위임 받을 수 있다. 위임역할들이 PBDM0와 PBDM1처럼 유사하게 보이지만, 위임역할의 소유주가 FBDR이지 사용자는 아니다. 때문에 TDBR은 역할 계층이 존재하지 않고 잘못된 권한위임은 발생하지 않는다. 그래서 역할-역할 위임이 유지 될 수 있다. 그림 4는 PBDM1과 PBDM2 각각의 역할 계층을 나타낸 것이다.[9]

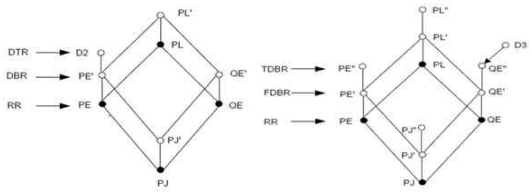

그림 4. PBDM1과 PBDM2에서의 역할과 역할 계층
fig 4. Role and Role hierarchy in PBDM1&PBDM2

2.4 ABDM에서의 권한위임

PBDM에서 확장된 ABDM은 사용자-사용자 권한으로 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.

A. 권한 부여 자격 - 역할은 필요한 자격을 갖춘 사람에게 지정되어야 하는 것처럼 위임 받을 능력도 특정 자격의 소유로 제한되어야 한다.
B. 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.
C. 위임 집합 - 어떤 조건에서 위임 될 권한의 집합을 정의한다.
D. 위임 단계 - 위임 정도에 따라 재위임이 일어날 때 기존의 위임 정도에 대한 관계를 명확히 정의한다[10].

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 좀 더 넓은 영역에서 접근제어를 구현하는 유연한 방법을 제공할 수 있다. 그러나 위임과 관련된 속성뿐 아니라 활성화, 범위 등 여러 부분에 대한 속성을 모두 유지하고, 그러한 속성간의 트리거를 통한 관계를 규정해야 하는 부담을 갖는다.

III. 제안 모델

본 논문에서는 역할 관리자 또는 위임자로부터 권한을 부여 받은 사용자가 직접 위임 역할을 생성하여 위임 할 수 있도록 하고, 사용자 부재 시 위임 권한의 영속성을 보장하며 위임의 폐지에서는 위임자와 관리자가 동시에 폐지의 권한을 가질 수 있을 뿐만 아니라 과도한 권한 위임이나 악의적인 위임에 대한 남용을 방지하기 위해 위임에 시간에 따른 제약을 줌으로써 위임의 사용을 제한 할 수 있는 TCPBDM을 제안한다.

TCPBDM에서 역할은 위임이 불가능한 고정역할 RR (Regular Role)과 다 단계 위임이 가능한 주기적 시간제한 역할 PCDR(Periodicity time Constraints Delegable Roles), 단순위임이 가능한 시간제한 역할 ACDR(Absolute time Constraints Delegable Roles), 위임 역할 DTR(Delegation Role)로 구분한다.

1. TCPBDM에서 시간 제약의 정의

TCPBDM에서 시간 제약은 주기적 시간제한과 기간적 시간제한으로 나뉜다. 주기적 시간제한은 일정한 간격으로 그 권한이 위임되는 것을 말하고 기간적 시간제한은 정해진 일정 기간 동안 사용자가 위임 받은 권한을 활용할 수 있다. 이런 시간제한을 활용하기 위해서는 첫째, 위임된 역할의 활성화된 시간제한과 둘째, 위임된 역할의 활성 시간 길이 제한, 셋째, 시간 범위 내에 활성에 대한 시간 길이가 제한되어야만 한다.

이에 TCPBDM에서는 시간의 범위를 시간이 시작하는 지점 T_{Start}와 그 시간이 끝나는 종점 T_{end}를 두고 그 사이에 활성화된 범위를 T_D로 정의한다. 이를 재정의 하면 다음과 같다.

[정의1] $T=\{(ts,te)|(ts,te)\in N\}$ 에 대해 $\forall s,e \in N$, $\forall ts,te\in T$, $s<e \Leftrightarrow ts< te$가 성립한다.

[정의2] 정의1에 의해 시간의 범위 $TD=\{(ts,te)|ts,te\in T\}$를 만족한다.

위 정의를 주기적 시간제한 위임 역할 PCDR과 기간적 시간제한 위임 역할 ACDR로 구분하고, PCDR의 시간제한 위임 같은 경우는 PCDR1[days-of-week hhmm, days-of

week hhmm]로 ACDR의 기간적 시간제한 위임의 경우는 ACDR1[Start time data, End time date]로 표기한다. 이를 표 1과 표 2로 정리하면 다음과 같이 두 가지로 나타낼 수 있다.

표 1. PCDR에서의 시간제한 활성 표현
Table 1. Time Constraints action expression in PCDR

PCDR		
Meaning	Predicate	Expression
위임1을 월~금까지,8:00am~6:00pm까지 활성	D1[Weekday 8:00 to 18:00]	PCDR1[Wd 800,Wd 1800]
위임2를 매주 8:00am~6:00pm까지 활성	D2[daily 8:00 to 18:00]	PCDR2[Da 800,Da 1800]
위임3을 월 8:00am~수 8:00pm까지 활성	D3[Monday 8:00 to Wednesday 20:00]	PCDR3[Mo 800,We 2000]
위임4를 주말 ~ 일요일 자정까지 활성	D4[Weekend 00:00 to 23:59]	PCDR4[Wk 0000,Wk 2359]

표 2. ACDR에서의 시간제한 활성 표현
Table 2. Time Constraints action expression in ACDR

ACDR		
Meaning	Predicate	Expression
위임1을 2010/03/01 8:00am~2010/03/02 6:00pm 활성	D1[Start 2010/03/01/8:00 to End 2010/03/02/18:00]]	ACDR1[6:00/1/Mar/2010, 18:00/2/Mar/2010]
위임2를 2010/04/01~2010/04/30 자정까지 활성	D2[Start 2010/04/01/00:00 to End 2010/04/30/23:59]	ACDR2[00:00/1/Apr/2010, 23:59/30/Apr/2010]
위임3을 2010/05/01~ 활성	D3[Start 2010/05/01 to NOT End]	ACDR2[00:00/1/May/2010,-]

표 1과 표 2를 보면 전자는 주기적으로 권한을 위임 받아 권한의 영속성을 보여주고 있고, 후자는 기간이 지나면 자동적으로 권한이 폐지됨을 볼 수 있다. 또한, 표 2에서 위임3과 같이 시작시간은 있으나 종점 시간이 없는 경우는 위임이 2010년 5월 1일부로 지속적인 권한 위임이 되었음을 알 수 있다. 그러나 ACDR의 경우는 단순위임 즉, 위임을 한번의 위임이 행해지면 위임 받은 사용자는 그 권한을 다른 이에게 위임 할 수 없으므로 이는 역할을 하나 생성한 것과 같은 효과라고 볼 수 있다.

2. TCPBDM 권한 위임 설계

그림 7과 그림 8은 TCPBDM에서의 위임 기법을 설명하기 위한 예이다. 그림 7에서 연산(Operations)과 객체(Object)는 수행될 시스템의 형태에 따라서 연산들이 달라진다. 그 예로 운영체제에서 연산자는 읽기(read), 쓰기(write), 실행(execute)이 될 것이고, DBMS라면 연산자는 삽입(insert), 삭제(delete), 추가(append), 갱신(update)이 된다. 예를 들어, 사용자 'John'은 'TL'의 역할을 담당하고 있다. 'TL'에는 'CS'와'CP' 객체에 대한 권한이 할당되어 있고, 'Tom'은 'PM'의 역할을 담당하고 'PM'에는 'CPP'객체에 대한 권한이 할당되어 있다. 'Smith'는 'PJ'의 역할로 'UP1B'객체에 대한 권한이 할당되어 있고, 'Alex'는 'QE'의 역할로 'John'처럼 'RvP'와 'ER'객체에 대한 권한 2개를 가지고 있다.

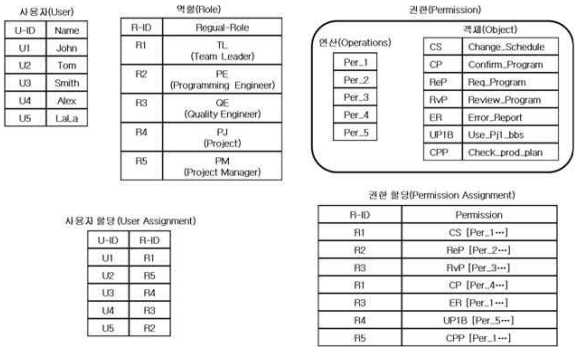

그림 7. 권한 위임의 경우
fig. 7. Case of permission delegation

위의 그림 7을 보면 다음과 같이 권한 위임을 위한 몇 가지 경우들을 말할 수 있다. Team Leader인 John이 Project manager인 LaLa에게 Confirm_Program권한에서 Per_4 만을 위임하고자 하는 경우와 Confirm_Program권한 전부를 시간제약을 가지고 LaLa에게만 일단계 위임을 하고자 하는 경우, 이럴 경우 LaLa는 John으로부터 위임 받은 권한은 다른 이에게 위임 할 수 없다. Jenny가 LaLa와 Smith에게 Change_Schedule 권한에서 Per_1을 LaLa에게 위임하고 Smith에게 권한 Per_2를 위임하고자 하는 경우, 그리고, John이 LaLa에게 Confirm_Program에서 권한 Per_4와 역할 Programming Engineer를 시간제약을 두고 단순 위임을 하고자 하는 경우들을 생각할 수 있다. 이중에 마지막 경우를 고려해보자.

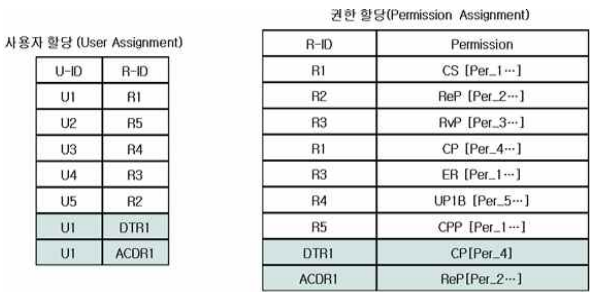

그림 8. TCPBDM에서의 위임 예
fig. 8. Example of Delegation in TCPBDM

TCPBDM에서의 역할 생성은 그 상위 계층에 제안한 모델을 추가함으로써 사용자가 위임할 권한의 집합으로 구성된 새로운 역할을 생성하고 위임 권한은 생성한 사용자 및 위임자가 소유권을 가지게 된다. 즉, D1과 D2라는 역할을 새로 생성하여 위임하고 하는 권한을 새로 할당한다. 이 때 새로 생성된 역할 D1은 일반적인 위임으로 다단계의 위임도 가능한 위임이다. D2는 시간적 제약을 가진 위임역할로써 단순 위임이다. 즉 D2에서 위임 받은 것을 다른 사용자에게 위임 할 수 없고 시간제한을 동반한다는 의미이다. 다음의 그림 8에서의 관리자 역할에 추가하고 위임하고자 하는 D1과 D2역할은 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 D1과 D2가 추가되는 되는 것은 그림 9에서 보는 것과 같다.

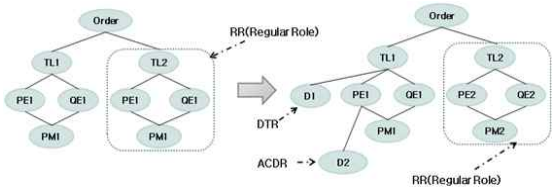

그림 9. 역할 계층 간의 변화
fig 9. The transition between the role hierarchy

그림 10은 본 논문에서 제안 하는 위임 기법을 나타내고 있다. 제안하는 기법은 위임하고자 하는 역할에 대해서 시간제한을 활용하여 위임자에게 역할과 권한을 할당하고 위임된 역할은 위임자와 관리자에 의해 위임과 폐지를 관여하게 된다. 또한, 시간제한으로 인하여 위임자나 관리자가 관여하지 않아도 시간 범위를 벗어나면 위임은 자동적으로 회수 또는 폐지가 이루어지게 되며, 위임 받은 역할에 대해 악용 할 경우 위임자가 원하는 시점에 회수를 할 수 있다.

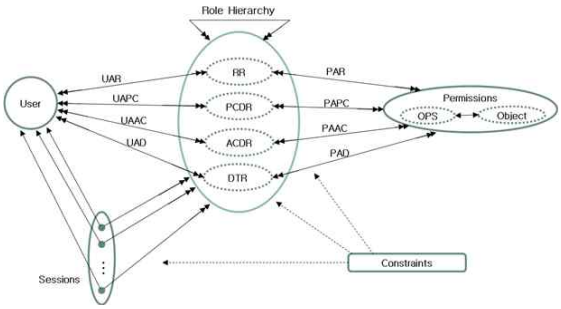

그림 10. 제안하는 TCPBDM
fig. 10. The proposed the TCPBDM

3. 모델간의 기본 집합과 함수정의

U: 사용자 집합 R: 역할 P: 권한 집합 RR: 일반 역할
DBR : 위임 할 수 있는 역할
PCDR: 주기적 시간제한 위임 역할
ACDR: 기간적 시간제한 단순위임 역할
DTR : 위임 역할 UAR: 사용자에 대한 역할 할당 관계
UAPC: 사용자에 대한 주기적 시간제한 위임 할당관계
UAAC: 사용자에 대한 기간적 시간제한 단순위임 할당관계
UAD: 사용자에 대한 위임 할당 관계
PA: 권한에 대한 할당 관계
PAD: 권한에 대한 위임 할당 관계
PAR: 권한에 대한 역할 할당 관계
PAPC: 주기적 시간제한 권한에 대한 역할 할당 관계
PAAC: 기간적 시간제한 권한에 대한 역할 할당 관계
제안 모델의 구조 및 정의는 다음과 같다.

[정의3] 실질적인 구성원의 역할은 RR, DBR, DTR로 구분되고, 다대다의 관계이다.

-UAR⊆ U×RR, UAPC⊆ U × PCDR, UAAC ⊆ U×ACDR
-DBR = PCDR ∨ ACDR , R = RR ∨ DBR ∨ DTR

[정의4] 위임 불가능한 역할 RR과 위임 가능한 역할들의 중복성이 없음을 나타낸다.

-RR ∧ PCDR = Ø , RR ∧ ACDR = Ø, RR ∧ DBR = Ø,
RR ∧DTR = Ø, DBR ∧ DTR = Ø

[정의5] 다단계 위임이 있을 때 UAB와 UAD로 위임 가능한 권한 집합을 사용자에게 사상하는 함수 관계

Permission*(u) = {p : P | ∃r ∈ DBR, (u, r) ∈ UAB ∧(r, p) ∈ PAB } ∨ {p : P | ∃r ∈ DTR, (u, r) ∈ UAD ∧ (r, p) ∈ PAD }

[정의6] PCDR은 주기적 시간제한 위임을 말하고, ACDR은 기간적 시간제한 위임들의 집합이다.

-PCDR⊆ DBR × TD (ts,te),td⊆TD -다중위임 가능
-ACDR⊆ DBR × TD (ts,te),td⊆TD -단순위임만 가능
-(PCDR ∨ ACDR) ∧ DTR =Ø, PCDR ∧ ACDR =Ø

[정의7] 각 권한 할당은 각 위임된 역할 지정된 권한 p를 모두 또는 일부 포함한다.

-PAR⊆ P×RR, PAPC⊆ P×PCDRn,
PAAC⊆ P×ACDRn, PA= PAR ∨ PAPC ∨ PAAC

[정의8] 고정 역할을 할당 받은 모든 구성원들은 위임 가능한 역할을 할당하고 있어야만 하고, 다른 구성원에게 자신의 고정 역할을 할당 할 수 없다.

-∀rr∈RR,∃u : U, pcdr : PCDR, acdr :ACDR ▪
(u,rr)∈URA ∧ rr = own_pc(pcdr) ∧
pcdr=own_ac(acdr)⇒user_r(rr)=user_pc(pcdr) ∧
user_pc(pcdr)=user_ac(acdr)

[정의9] 위임 역할 집합을 위임 가능한 주기적 시간제한 역할에 사상한다.

-own_d(r): PCDR→$2^{DTR}$and
(pcdr1,pcdr2∈PCDR,dtr∈DTR)▪(pcdr1≠pcdr2)∧(dtr∈own_d(pcdr1)∧dtr∈own_d(pcdr2))

이와 같이 [정의4]는 상속 관계를 정의하여 역할의 중복을 방지한다. [정의6]는 [정의1]과 [정의2]에 의해 시간적 제한을 활용하여 위임하게 된다. 또한 위의 모든 정의의 기본 형식은 일반적인 RBAC의 사용자와 역할 그리고 역할과 역할의 관계를 나타낸 정의와 형식에 따른다.

4. 지정된 권한만의 위임

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

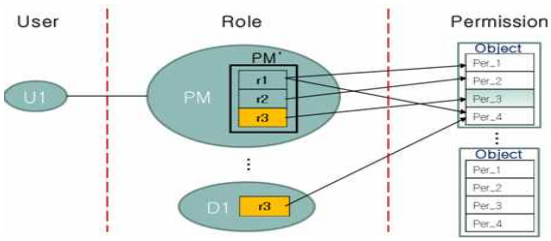

그림 11. 변경된 의미의 역할 위임
fig. 11. Changed the meaning of role delegation

그런 이유로 제안된 모델에서의 역할은 업무의 집합을 의미한다. 즉 역할과 관련된 작업의 세분화를 통해서 권한의 일부분을 위임 할 수 있도록 한다. 그림 11에서 PM의 작업 r1(설계), r2(분석), r3(구현)으로 나눌 수 있고, 관리자로부터 권한을 부여 받은 U1은 r3에 관한 권한만을 제3자에게 위임하기를 원한다고 가정한다. 변경된 의미의 역할에서는 PM을 작업의 단위로 세분화했기 때문에 원하는 위임 역할을 생성하여 자신의 권한인 업무 r3만을 선택해서 위임 할 수 있다.

5. 위임 감독

이제까지 제안한 모델의 상위 계층에서는 위임될 대상에 대해 권한을 부여 받은 역할과 기존의 역할이 갖는 제약 조건을 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 그 역할의 행위가 위임자에 의해 감독되어야 하며, 같은 역할 계층에서 일어난 위임도 이러한 위임이 적합한가에 대한 동일 범위 여부에 대한 판단을 해야 한다.

위임 역할에서 ACDR은 기간적 시간제한 단순 위임으로 생성된 역할이기 때문에 다른 역할과의 계층관계를 갖지 않는다. 따라서 그러한 역할을 감독할 만한 역할이 존재 하지 않고, 제한 받은 시간 범위를 소비하면 자동으로 권한이 폐지된다. 또한, 다른 역할과의 계층관계를 갖지 않는다는 점을 보안하기 위해 위임 역할에 대한 감독을 위임자뿐만 아니라 관리자도 같이 위임 역할을 생성한 역할이 위치한 역할 계층에서 상위 역할이 가지는 감독, 권한에 대한 정보를 상속 받는다. 따라서 새로 생성된 ACDR 위임역할에 대한 감독은 기존 역할에 대한 감독을 하고 역할 계층상의 상위 역할이 위임역할에 대한 감독도 한다. PCDR위임 역할을 할당을 한 경우, 이는 정적 할당이 일어난 것이다. 권한을 소유한 역할에 위임 역할을 활성화 하여 해당하는 허가를 이행하려면, 세션 내에서 해당 역할을 활성화하기 위해 상위 감독 역할에 의해서 정적 역할-역할 할당 후에서야 가능하다. 이것 없이 위임 역할에 대한 할당이 일어나면 실제 할당된 그 역할을 세션 내에서 활성화 시키지 못하여 작업을 할 수 없다. 즉 위임 역할을 이행 할 수 없게 된다.

IV. 비교 분석

다음 표3은 기존의 RBAC96과 역할위임 모델인 RBDM0과 권한 위임 모델인 PBDM0와 PBDM1을 제안 모델인 TCPBDM을 비교 분석하였다. 비교는 위임에 관련된 6개의 각 항목별로 비교하였으며, 의무 분리와 권한 분리에 모델이 각 항목을 지원하지의 여부를 비교 분석하였다[14]. 또한 보안측면에서는 공통 평가 기준에서 명시하는 보안 기능 요구사항을 기준으로 본 모델을 평가하였다.

표 3. 기존 모델들과 제안 모델의 위임비교
Table 3. Compare existing models and the proposed model

비교기준		RBAC96	RBDM0	PBDM0	PBDM1	TCPBDM
위임	위임 유형	지원하지 않음	단순위임	단순 위임	단순위임 다단계위임	단순위임 다단계 위임
	부분 권한위임	지원하지 않음	역할 부분집합	역할 부분 집합	권한 부분 집합	권한 부분 집합
	위임 개부	지원하지 않음	지원하지 않음	지원하지 않음	지원하지 않음	피 위임자
	위임 역할 회수	지원하지 않음	관리자 사용자	보안관리자 사용자	보안관리자 사용자	위임자 관리자 시간제한
	시간제한적 위임	지원하지 않음	지원하지 않음	지원하지 않음	지원하지 않음	**PCDR** **ACDR**
	위임 관점	지원하지 않음	사용자	사용자	사용자	역할 시간제한

위임 유형 항목에 대해서는 역할 기반 모델의 초기 모델인 RBAC96은 위임을 고려하지 않았고, RBDM0와 PBDM0은 단순 위임만을 지원하여 언제든 회수가 가능한 위임을 고려하였다. 또한, PBDM1은 단순위임과 다단계 위임을 동시에 지원한다. TCPBDM은 ACDR과 PCDR로 인해 전자는 단순위임을 후자는 다단계위임을 지원함을 보였다.

부분 권한위임은 자신의 역할에 할당된 권한을 일부만 위임하는 것으로 RBAC96은 위임이기 보다는 상속의 개념이고, RBDM0은 역할을 기반으로 위임하여 자신의 권한 전부를 위임 하는 형태를 취하고 있으며, PBDM0은 RR을 생성하여 사용자에게 위임하는 형태로 권한이 역할의 부분집합으로 속해 있고, PBDM1은 위임을 생성할 시에 전제조건 상태와 위임 범위, 최대 위임 깊이로 위임 할 수 있는 조건하에서 권한에 대한 부분을 위임하는 부분권한 위임을 취하고 있다. TCPBDM은 권한을 세분화 하여 자신의 속한 권한을 새로 생성된 역할에 할당하여 위임한다. 이는 권한을 부분으로 나누어 권한 집합 내의 위임을 뜻하는 말이기도 하다.

위임 역할 회수은 위임 역할이 새로 생성된 역할에 대한권한을 계속 유지함으로서 새로 생성된 역할의 부정행위가 발생되는 경우나 새로 생성된 역할에 대한 역할 남용이 발생되는 경우 등 위임 역할이 원하지 않는 경우에는 언제든 회수가 가능한 경우를 나태는 것으로, 새로 생성된 역할에 위임 역할의 권한이 복사되어 위임되는 경우와 위임 역할의 권한도 같이 생성된 역할에 위임 되는 경우로 볼 수 있다. 기존의 모델들은 모두 관리자나 보안 관리자가 역할을 회수 하는 방식이고, TCPBDM은 위임 역할이 새로 생성된 역할의 부정행위를 방지하기 위해 역할 감시를 하게 되고 동시에 관리자나 위임 역할 없이도 시간적 제한에 의해 시간의 범위가 초가하면 자동으로 회수되는 방식이다.

위임 관점은 위임하는 대상이 사용자 입장에서 위임 하는 것 인지 역할 입장에서 하는 것인지를 나타내며, 기존의 모델들은 모두 사용자 입장에서 위임을 했으며, 실세계에서는 사용자-사용자 위임은 거의 이루어지지 않고 역할-역할의 위임이 이루어지고 있다. 다른 모델과 다르게 TCPBDM은 역할 입장에서 역할-역할 위임 방식이다.

표 4는 보안에 있어서 모델 및 시스템이 가져야 할 보안 요소들 중에 하나 인 무결성을 3개의 각 항목별로 기존의 모델들과 비교하였으며, 업무 분리와 권한 분리는 보안 시스템에 의해 관리되는 정보의 무결성 보장을 위해 정보의 무결성에 영향을 미치는 역할들에 배정되는 역할들을 통제함으로써 보안 시스템에 의해 관리되는 보안 특성을 유지한다. 업무분리는 RBAC2와 TCPBDM만이 지원하는 반면 권한 분리는 TCPBDM이 지원함을 보였다.

표 4. 기존 RBAC모델과 제안 모델의 무결성 비교
Table 4. Compare RBAC Model and The proposed Model of integrity

비교기준		RBAC0	RBDM1	RBAC2	TCPBDM
무결성	임무 분리	지원하지 않음	지원하지 않음	**SSD** **DSD**	**SSD** **DSD**
	권한 분리	지원하지 않음	지원하지 않음	지원하지 않음	**SSP** **DSP**
	사용자수	최대 사용자수제한	최대 사용자수제한	최대 사용자수제한	최대 위임자수 제한

V. 결론 및 향후 연구 과제

본 논문에서는 기존의 RBAC 표준 참조 모델을 기반으로 위임 기법에 시간제한을 추가하여 모델링 하여 보았다. 즉, 권한 위임에 중점을 두어 역할 관리자는 모든 역할과 권한을 관리할 수 있는 역할을 가지며, 각 역할은 자신의 권한을 제3자에게 위임을 하고자 하는 경우에는 역할 관리자의 역할에서 위임자가 위임하고자 하는 역할을 복제하여 위임을 하게 된다. 또한 위임의 폐지에서는 위임자와 역할관리자가 동시에 폐지의 권한을 가지게 되고, 시간적 제한을 사용하여 적용된 시간 범위 내에서 권한을 활성화하게 되며 그 시간 범위를 벗어나면 위임자나 역할관리자가 관여하지 않아도 자동으로 폐지되는 위임 기법을 제안하였다.

본 논문에서의 위임 기법을 기존의 모델과 비교 분석하여 RBAC을 바탕으로 부분 권한 위임이 추가됨과 임무 분리와 권한 분리가 지원됨을 보였다. 향후 연구과제로는 위임을 시간 제한적으로 보다는 공간제한적인 의미에서 접근하여 접근제어 모델을 연구 할 것이며, 비교분석에서는 위임기법이 적용된 다른 모델들과 비교하고자 한다.

참고문헌

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

[2] 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, 2003.

[3] Ezedin Barka and Ravi Sanhu, "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.

[4] Ezedin Barka and Ravi S Shanhu, "A Role-Based Delegation model and Some Extensions," Proc, Of 23rd National Information System Security Conference(NISSC2000), pp. 168-176, Dec., 2000.

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

[6] Zhang L, Ahn .G.J 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.

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

[8] Chunxiao Ye, Yunqing Fu, Zhingfu Wu, "An attribute-Based-Delegation-Model," ACM International Confer ence Proceeding Series, Vol. 85, Proceedings of the 3rd international Conference in Information security, pp. 220-221, November 14-16, 2004.

[9] 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.

[10] J.B.D. Joshi, E. Bertino. U. Latif, A. Gahfoor. Gemeralized, "A generalized Temporal Role Based Access Control Model," IEEE Transactions on Knowledge and Data Engineering, 17 (1), pp. 4-23, Jan., 2005 .

[11] J.B.D. joshi, E. Bertino."Fine-grained role-based delegation in presence of the hybrid role hierarchy," Proc. 11th ACM, Access control model and Technologies. pp. 81-90, Jun., 2006.

[12] J. Wainer and A. Kumar, "A Fine-Grained, Controllable, User-to-User Delegation Method in RBAC," Proc. 10th ACM Symp. Access Control Models and Technologies(SACMAT '05), pp. 59-66, June 2005.

[13] Y. Zhang. "Achieving Flexible Task Delegation in Role-Based Agent Teams," Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 2007, pp. 3801-3806, 7-10 Oct. 2007.

[14] Hagstrom, A, Jajodia, S, Parisi-Presicce, F, Wijesekera, D,"Revocation-a Classification," Computer Security Foundations Workshop, 2001. Proceedings. 14th IEEE, pp. 44-58, Oakland, May 7-9, 2001.

저 자 소 개

김 태 식

2003 : 동국대학교 컴퓨터멀티미디어공학 공학사

2006 : 동국대학교 컴퓨터공학과 공학석사

2006 - 현재 : 동국대학교 컴퓨터공학과 박사수료

관심분야 : 정보 보안, 컴퓨터네트워크, 유비쿼터스 컴퓨팅

장 태 무

1977 : 서울대 전자공학과 공학사

1979 : 한국과학기술원 전산학과 공학석사

1995 : 서울대 컴퓨터공학과 공학박사

1981 - 현재 : 동국대학교 공과대학 IT학부컴퓨터공학과 교수

관심분야 : 병렬처리컴퓨터, 컴퓨터구조, 보안, 유비쿼터스 컴퓨팅