JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 27 No. 2, pp. 107-114, February 2022
https://doi.org/10.9708/jksci.2022.27.02.107

APDM : Adding Attributes to Permission-Based Delegation Model

Si-Myeong Kim*, Sang-Hoon Han**

*Adjunct Professor, Department of Computer Science and Engineering, Dongguk University, Seoul, Korea
**Professor, Dept. of Computer Information Security, Korea National University of Welfare, PyeongTaek, Korea

[Abstract]

Delegation is a powerful mechanism that allocates access rights to users to provide flexible and dynamic access control decisions. It is also particularly useful in a distributed environment. Among the representative delegation models, the RBDM0 and RDM2000 models are role delegation as the user to user delegation. However, In RBAC, the concept of inheritance of the role class is not well harmonized with the management rules of the actual corporate organization. In this paper, we propose an Adding Attributes on Permission-Based Delegation Model (ABDM) that guarantees the permanence of delegated permissions. It does not violate the separation of duty and security principle of least privilege. ABDM based on RBAC model, supports both the role to role and user to user delegation with an attribute. whenever the delegator wants the permission can be withdrawn, and A delegator can give permission to a delegatee.

▸ **Key words**: RBAC, Role, Delegation, Attibute, Permission

[요 약]

위임은 사용자에게 접근 권한을 사상(mapping)하여 유연하고 동적인 접근제어 결정을 제공하는 강력한 메카니즘이다. 또한, 분산환경에 유용하다. 대표적인 위임 모델 중 RBDM0 와 RDM200은 사용자 대 사용자 위임으로서 역할위임이다. 그러나, RBAC에서 역할계층의 상속 개념은 실 기업 조직의 관리 규칙과 조화롭지 못하다. 본 논문에서 우리는 위임 권한의 영속성을 보장하고 의무 분리 원칙과 최소 권한의 보안 원칙에 위배되지 않는 속성을 첨가한 권한위임 모델 (APDM)을 제안한다. RBAC모델을 기반으로 하는 APDM은 속성을 사용하여 역할 대 역할과 사용자 대 사용자의 위임을 제공한다. 위임자는 원하는 권한만을 특정인에게 위임할 수 있고, 속성을 활용하여 위임자가 원하는 시점에서 권한을 회수할 수 있다.

▸ **주제어**: 역할기반, 역할, 위임, 속성, 권한

• First Author: Si-Myeong Kim, Corresponding Author: Sang-Hoon Han
*Si-Myeong Kim (creta72@dongguk.edu), Department of Computer Science and Engineering, Dongguk University
**Sang-Hoon Han (shhan@knuw.ac.kr), Dept. of Computer Information Security, Korea National University of Welfare
• Received: 2022. 01. 25, Revised: 2022. 02. 14, Accepted: 2022. 02. 14.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

인터넷과 네트워크가 발달함에 따라 정보의 공유가 증가하고 그 속도는 빨라졌으며 종류 또한 다양하다. 이를 해결하기 위한 방안으로 접근제어(access control)의 필요성이 높아지고 있다[1]. 접근제어는 임의적 접근제어(DAC)와 강제적 접근제어(MAC) 두 가지로 나눌 수 있다. 대부분 시스템에서 접근제어의 관리를 위해 사용자를 집단으로 묶는다. 권한 집합과 사용자 집단을 대표는 개념에서 역할을 정의하고, 그런 역할로 접근제어를 수행하는 역할기반 접근제어(RBAC:Role-Based Access Control)가 있다[1][2].

속성기반 접근제어(ABAC)는 주체와 객체의 속성에 대한 조건을 기술해 어떤 객체에 접근하기 위해 만족하는 속성을 정의하고, 그 객체에 접근하는 주체의 속성이 있는지 여부를 검사하는 접근제어이다[8][9][10][11].

위임은 유연하고 동적인 접근제어를 제공하는 강력한 메커니즘이다. 또한, 분산환경에서 특히 유용하다. RBAC모델은 사용자의 역할 위임에 관하여 하지 않고 관리 측면에서의 위임만을 정의한다[2][3]. 또한, 사용자가 사용할 수 있는 업무의 권한을 선정하고 선정된 권한 외에는 가질 수 없도록 하는 접근제어를 제공하고 역할의 계층을 통해 역할이 갖는 권한을 설정하고 역할 간의 충돌을 사전에 방지한다[3][4][5], 위임은 단순과 다단계 위임 두 가지 방법이 존재한다. 단순위임은 위임받은 역할을 제 3자에게 자신이 위임할 수 없다는 것을 의미하고, 다단계 위임은 자신이 위임 받은 역할을 다시 제 3자에게 위임할 수 있다는 것을 의미한다[6][7]. 위임을 대표하는 모델이 RBDM(Role-Based Delegation Model), PBDM(Permission-Based Delegation Model), ABDM(Attribute-Based Delegation Model) 등이 있다. 그러나 어떤 방법을 사용해도 RBAC 특성상 단순 역할만을 위임할 경우 피위임자에게 과도한 권한이 위임되게 된다. 또한, 사용자는 단지 역할과 관계를 갖기만 할 뿐 권한과 직접적인 관계 유지를 하지 않는다. 그래서 위임받은 권한에 대한 관리 감독하는 측면에서도 효율적이지 못하다. 그리고, 권한 분배의 경우 권한위임이 발생했을 때 역할 상속에 의한 지속적인 권한위임을 유지해야 하는 문제가 있다[3][5][7][8].

따라서, 본 논문에서는 역할 업무에 따라 직접 사용자가 역할을 통하여 권한위임을 할 수 있고, 위임 시 제 3자에게 발생 되는 과도한 위임이나 악의적인 위임을 방지하며 효율적인 관리가 가능한 속성 권한위임 모델을 제안한다.

본 논문의 구성은 1장에서 제안 모델의 관련 연구와 필요성을 제시한다. 2장에서는 RBAC 모델과 위임 기법 및 각 모델의 위임에 대해 논하고, 3장에서는 APDB 속성기반의 권한위임 모델을 제안하고, 기존의 모델과 제안하는 모델을 분석한다, 5장에서는 결론으로 끝을 맺는다.

II. Preliminaries

1. RBAC

RBAC모델의 구조는 RBAC0(Flat RBAC), RBAC1(Hierarchical RBAC), RBAC2 (Constrained RBAC), RBAC3(Symmetric RBAC)로 나뉘고, 하위 단계에서 상위 단계로 올라가면서 하위 단계의 특징들을 내포한다[1][2].

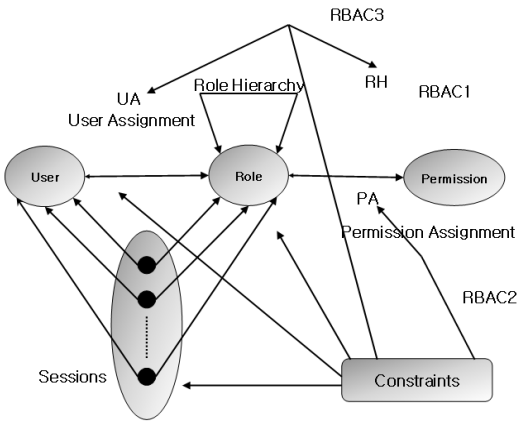

Fig. 1. RBAC Model

그림 1은 RBAC모델간의 관계를 나타낸 것이다. RBAC의 구성 요소는 사용자, 역할, 권한이 있다. 사용자는 사람 또는 프로세서로 볼 수 있고, 역할은 그 사용자에게 사상된 권한을 가진 조직 내의 조직책임과 직무 이름이다. 권한은 권한을 사상받은 사용자에게 시스템에서 특정한 이행을 할 수 있는 수단이다.

RBAC0는 사용자 대 역할 사상과 권한 대 역할 사상이 다대다 관계를 나타낸다. RBAC1은 RBAC0의 구성에 역할 계층이 추가된 개념이다. 역할계층은 조직 내에서 책임의 순서와 권한을 반영하기 위해 역할을 구조화하는 것이다. RBAC2는 그림1에서와 같이 RBAC0에 제약을 두는 구조로 제약은 사용자 대 역할 사상과 사용자 대 세션 연결로 역할들의 촉진이 관련된다. RBAC3은 RBAC2에 권한 대 역할 간의 요구사항을 추가한 것이다[3][4][5].

2. Delegation

위임은 권한과 기타 속성을 사용자에게 사상하는 메커니즘이다. 위임을 수행하는 사용자를 “delegator”라하고, 해당 사용자를 위임받은 사용자를“delegatee”라 한다.

권한 속성을 성공적으로 제공하거나 사용자로부터 다른 사용자에게 전달할 수 있는 경우 “delegable”이 된다[6][7][8][9]. 위임 발생 상황은 다음과 같이 세 가지로 볼 수가 있다.

첫째, 사용자가 업무에 지장이 없도록 다른 제 3자에게 자신의 역할 일부를 수행하게 하는 백업을 생성하는 경우이다. 둘째, 작업의 효율을 위해서거나 조직을 구성해서 한 사람 또는 그 부서에 사상된 권한을 다른 이에게 다시 분배하는 경우이다[7][8][9]. 셋째는 서비스를 받기 위해 원격 제어 호출을 하는 경우이다. 호출한 주체로 권한을 실행하기 위해 주체에 대한 권한 방법을 실행하는 사용자에게 제공하는 개념으로서 위임된다[6][7].

위임은 역할위임, 권한위임, 속성위임에 따라 역할을 기반으로 하는 위임 모델은 RBDM0와 RDM2000이 있고, 권한을 기반하는 PBDM, 속성을 기반하는 ABDM등으로 분류된다. 이들이 RBAC모델을 기반으로 하는 가장 대표적인 위임 모델이다.

2.1 Delegation in RBAC

RBAC에서 역할계층의 상속 개념은 상속을 통하여 하위 역할이 받은 권한을 상위 역할 자신이 이행할 수 있도록 권한 위임하였어도, 상속에 의해 위임된 권한이 지속적으로 유지된다는 문제가 발생한다. 이는 실제 기업 조직의 관리와는 잘 어울리지 않는다. 이를 방지하기 위해 역할계층 상에서 상속에 대한 한정 관리를 해야 한다. 관리 감독과 같이 한정된 권한에 대한 상속을 허용하고, 상위 역할로 상속 허용을 하여 상위 역할이 해당 역할을 이행할 수 있도록 한다. 단순하게 상위 역할이 하위 역할의 백업이 된다는 것이 문제다.

2.2 Delegation in RDM2000

RBAC모델 중 RBAC0에 기반하며 가장 간단한 형태이다. RBDM에서의 위임은 역할 상속이 이루어지지 않은 형태로 사용자들 사이에서 이루어진다. 그 가정과 기본 요소를 보면 같은 역할을 가진 사용자들은 위임이 허용되지 않고 단순 단계 위임만이 가능하다. 이것은 위임된 역할이 제 3자에게 위임될 수 없다는 것을 의미하며 본 구성원만이 위임 가능함을 보여준다. 이러한 위임은 전체적 위임, 즉 위임할 수 있는 역할을 가지고 있는 개별 사용자는 그 역할에 포함된 권한 전부를 위임하거나 전혀 위임하지 않는다. 이러한 RBDM을 기반으로 확장된 RDM2000은 그림 2과 같은 Depth로 위임경로의 깊이를 나타낸다.

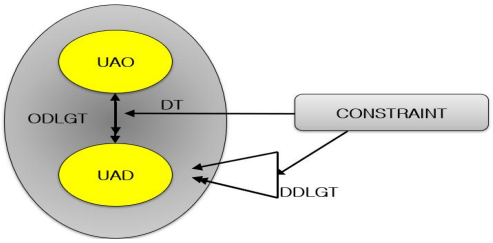

Fig. 2. Delegation relationship in RDM2000

RDM2000의 사용자-사용자 위임에 대해 위임 역할, 위임사용자, 위임된 역할, 위임된 사용자들의 구성 요소 사이의 관계를 예로 들면 그림 3과 같다.

즉 (Linda,PL1,Tony,Q2)의 의미는 Linda 역할 PL1을 촉진하여 Tony에게 역할 Q2를 위임한다. 위임 관계는 원 사용자 위임과 위임된 사용자 위임으로 분류된다. 이를 기반으로 함수를 정의한다. 예를 들면 함수 Alice가 UA(u2,r2)을 다른 UA(u3,r3) 혹은 ø에 사상하기 위해 그리고, Path 함수는 위임경로를 UA로 사상한다. Depth 함수는 위임경로를 반환한다. 위임경로는 순서화된 사용자-역할 배치 관계 집합이며, 다단계위임이 발생 될 때 위임경로가 생성된다. 위임 관계는 다음의 집합을 갖는다.

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

DLGT	Delegation Path
D1	P1: (Linda, PL1),(Lejk, DIR)∈DLGT
D2	P2: (Alice, PE1),(Linda, PL1)∈DLGT
D3	P3: (Jadden, PE1),(Linda, PL1),(Lejk, DIR)∈DLGT
D4	P4: (Tony, QE2),(Lejk,DIR)

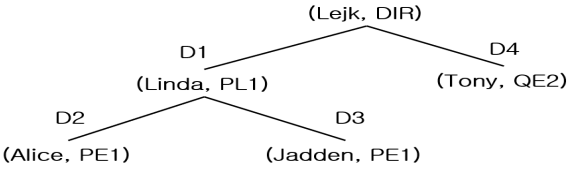

Fig. 3. An example for Delegation Paths and and Delegation Tree.

위의 위임에서 Path 함수를 통해 위임경로가 P1, P2, P3, P4임을 알 수 있다. 경로 관계는 그림 3에서 보여주고 있다[6].

계층 구조를 한정하기 위해 위임트리의 깊이와 넓이의 한계를 부과하여 결정한다. 이러한 위임의 깊이를 제어하는데 최대 깊이를 결정할 수 있으나, 위임 관계의 넓이를 결정할 수 없다는 단점이 있다.

2.3 Delegation in PBDM

PBDM은 RDM2000을 기반으로 확장하여 사용자 대 사용자 권한을 위임하는 모델로써 PBDM0, PBDM1, PBDM2로 나뉜다. 사용자가 권한위임을 다른 이가 대행하기 위해 임시 역할인 RR을 만들고 위임할 권한을 사상하여 전달한다. RR에 권한을 사상하여 전달하기 때문에 위임 해제 시에는 RR 자체를 삭제하거나 사상된 권한 중 일부만을 삭제하면 위임을 회수할 수 있어 효율적이다. 그렇지만, 관리자 없이 사용자에 의해 생성된 역할은 관리가 되지 않아 나쁜 의도로 권한을 위임할 수 있다. 이 단점을 보완하기 위한 것이 PBDM1이다. PBDM1은 PBDM0을 기반으로 권한위임이 가능한 역할 DBR(Delegable Role)과 불가능한 역할 RR로 역할을 나누어 확장한 모델이다.

사용자는 권한을 위임하기 위해 역할 DTR을 생성한 후, 역할 DBR에 사상된 권한 중 위임할 권한을 DTR에 사상하여 위임한다. 이런 위임을 위해 만들어진 DTR은 만든 사용자가 직접 관리하게 된다.

PBDM2는 PBDM1을 확장한 모델로써 역할 대 역할 위임으로 PBDM0과 같이 권한을 위임 불가능한 권한과 위임 가능한 권한으로 구성한다. RR에 사상된 권한은 위임이 불가능하고, FDBR에 사상된 권한은 위임이 가능하다. TDBR은 임시 위임 역할으로써 역할-역할 사상을 가진 위임자로부터 권한위임을 부여받을 수 있고, FBDR에 의해 권한위임을 부여받을 수 있다. 위임 역할들이 PBDM0와 PBDM1처럼 유사하게 보이지만, 위임 역할의 소유주가 FBDR이지 사용자는 아니다. 그래서 TDBR은 역할계층이 없고 악의적이거나 남용되는 권한위임이 없다. 그렇기에 역할-역할 위임 유지가 가능하다. 그림 4는 PBDM1과 PBDM2에서의 역할계층을 나타낸 것이다[8][9][10].

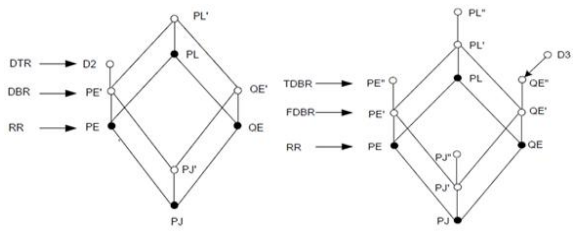

Fig. 4. Role and Role hierarchy in PBDM1&PBDM2

2.4 Delegataion in ABDM

ABDM은 사용자 대 사용자 권한으로 위임하기 위해 여러 조건 또는, 한정을 역할과 관련된 속성으로 정의하여 위임을 구현한 방법이다. 위임 역할을 위한 완연한 접근제어 방법을 위해 다음 사항을 역할의 속성으로 함유한다.

A. 권한 제공 자격 - 역할은 필요한 자격을 가진 사람에게 지목되어야 하는 것처럼 위임받는 것도 특정 자격의 소유로 한정돼야 한다.
B. 권한 제공 조건 - 역할 소유자가 권한에 대한 책임과 위임받을 사람의 자격이 없는 상황의 경우 권한 제공 조건이 있을 수 있다.
C. 위임 집합 - 어떤 조건에 따라 위임될 권한 집합들로 정의하고 있다.
D. 위임 단계- 위임 수준에 따라 다시 위임이 발생할 때 기존 위임 수준에 관한 관계를 명백히 정의한다[10].

이러한 속성은 일반적인 손실이 없이 정책에서 구현될 수 있고, 역할 속성으로 정의하여 구현할 경우가 역할 외부 정책으로 구현하는 것보다, 분산환경에서 접근제어를 구현하는 방법을 제공할 수 있다. 이런 제약은 편리하고 관리가 쉽게 기술돼야 한다. 그러나 위임과 관련된 속성, 촉진, 범위 등을 모두 유지하고, 그러한 속성 간의 연쇄작용을 통한 관계 규정을 해야 하는 부담을 갖는다.

III. The Proposed Scheme

기존의 RBAC모델에서의 위임에 관한 연구는 주로 관리자의 업무와 분산 시스템 환경에서의 위임에 초점을 맞추고 있었다. 하지만 사용자가 원하는 위임과 위임된 역할에서의 제 위임에 대한 관리, 잘못된 위임에 대한 흐름, 악의적인 위임에 대한 관리가 필요하다.

이 장에서는 위임자 또는 관리자로부터 권한을 위임받은 사용자가 직접 다른 이에게 위임할 수 있도록 하고, 갑자기 사용자 부재가 발생했을 시 위임 권한의 지속성을 보장하며 위임 삭제는 관리자와 위임자가 함께 관리할 수 있고, 악의적인 위임이나 과도한 위임 남용을 방지하기 위해 속성에 따른 위임 제약을 함으로써 위임 사용을 한정할 수 있는 APDM을 제안한다.

APDM에서 역할은 위임이 불가능한 역할 RR(Regular Role)과 위임 역할 DTR(Delegation Role), 다단계위임이 가능한 비속성 중심역할 NADR(non Attribute-centric Delegtable Roles), 단순위임만이 가능한 속성 중심역할 ADR(Attribute-centric Delegatable Roles)로 구분한다. 위임자는 자신의 권한 모두 또는 일부를 위임할 수 있다. 속성은 [11][12][13]에서 제안한 속성 표현을 기반으로 활용했다.

1. Formal Definition of The Proposed model

본 논문에서 제안하는 APDM에서 속성은 비속성 중심 역할과 속성 중심역할로 구성된다. 다단계 비속성 집합은 다단계위임이 가능하며 특정 위임받은 권한을 활용할 수 있고, 속성 집합은 단일 위임만 가능하며 특정 권한을 활용할 수 있다[13][14].

이런 속성을 활용하기 위해 첫째, 위임 역할의 촉진된 시점, 둘째 위임 역할의 촉진 한정이다.[13][14][15]

[정의] 1. 위임 데이터 함수

Ab:위임이 발효되는 데이터

Ae:위임이 만료되는 데이터

[Ab,Ae]:위임이 발효돼서 만료 기간의 A 속성.

A[A-atr]=A기간=[Ab,Ae] : 유효 위임 속성

[정의] 2. 위임 기간 함수

ADb : 위임된 역할이 효력을 발생하는 시점

ADe : 위임된 역할이 만료되는 시점

[ADb,ADe]:위임된 역할이 발효돼서 만료된 AD속성.

AD[AD-art]=AD기간=[ADb,ADe]:유효 위임기간 속성

[정의] 3. 속성 함수

[Ab, Ae]∧[ADb,ADe] : A≦Ab

[A,Ae]∧[ADb,ADe] : Ab<A<Ae∧AD≦ADe

[A,A2]∧[AD,ADe] : Ab<A<Ae∧ADb<AD<ADe

∅ : Ab<A<Ae∧ADe≦AD∨Ae≦A

2. Design of Permission Delegation in APDM

본 논문에서 그림 5과 그림 6은 APDM에서의 위임 기법을 설명하기 위한 예이다. 그림 5에서 객체(Object)와 연산(Operations)은 실행될 시스템의 형태에 따라서 연산들이 달라진다. 그 예로 운영체제에서 연산자는 읽기, 쓰기, 실행이고, 데이터베이스라면 연산자는 삽입, 삭제, 추가, 갱신이 된다. 예를 들어, 사용자 `Jadden'는 `PM`의 역할이다. 'PM'에는 'CPP'에 대한 권한이 사상되어 있다. 'Tom'은 'QE'의 역할로 'Alex'가 'CS'와 "CPP'에 대한 권한을 가지고 있는 것처럼 'RvP'와 'ER'에 대한 권한을 가지고 있다.

그림 5를 보면 다음과 같이 위임을 위한 몇 가지 경우가 있다. Team Leader인 Alex가 Project manager인 'LaLa'에게 CP 권한에서 Per_3만을 위임하고자 하는 경우와 CP 권한 모두를 속성을 가지고 'LaLa'에게만 단순위임을 하고자 하는 경우 'LaLa'는 'Alex'으로부터 위임받은 권한을 제3자에게 위임할 수 없다. 'Alex'가 'LaLa'에게 CP에서 권한 Per_4와 역할 PE에 속성을 두고 단순위임을 하고자 하는 경우를 고려해보자.

User

U-ID	Name
U1	Alex
U2	Jadden
U3	Smith
U4	Tom
U5	LaLa

Role

R-ID	Regual-Role
R1	TL (Team Leader)
R2	PE (Programming Engineer)
R3	QE (Quality Engineer)
R4	PJ (Project)
R5	PM (Project Manager)

Permission

Operations	Object	
Per_1	CS	Change_Schedule
Per_2	CP	Confirm_Program
Per_3	ReP	Req_Program
Per_4	RvP	Review_Program
Per_5	ER	Error_Report
	UP1B	Use_Pj1_bbs
	CPP	Check_prod_plan

User Assignment

U-ID	R-ID
U1	R1
U2	R5
U3	R4
U4	R3
U5	R2

Permission Assignment

R-ID	Permission
R1	CS [Per_1…]
R2	ReP [Per_2…]
R3	RvP [Per_3…]
R1	CP [Per_4…]
R3	ER [Per_1…]
R4	UP1B [Per_5…]
R5	CPP [Per_1…]

Fig. 5. case of Delegation

APDM에서의 역할을 만든다는 것은 상위 계층에서 제안한 모델을 추가하여 사용자가 위임할 권한들의 집합으로 구성되어 새로운 역할 만들고 위임에 따른 권한은 새롭게 만든 사용자와 위임자가 소유권을 가지게 된다. 이때 새로 만들어진 역할 R1은 다단계의 위임도 가능하다. R2는 속성 중심을 가진 위임 역할로써 단순위임이다. 다음의 그림 5에서 관리자 역할에 역할을 추가하고 위임하고자 하는 R1와 R2 역할은 복제하여 위임자와 관리자가 위임을 주관하는 권한으로 사상한다.

User Assignment

U-ID	R-ID
U1	R1
U2	R5
U3	R4
U4	R3
U5	R2
U1	DTR1
U1	ADR1

Permission Assignment

R-ID	Permission
R1	CS [Per_1…]
R2	ReP [Per_2…]
R3	RvP [Per_3…]
R1	CP [Per_4…]
R3	ER [Per_1…]
R4	UP1B [Per_5…]
R5	CPP [Per_1…]
DTR1	CP[Per_4]
ADR1	ReP[Per_2…]

Fig. 6. Example of Delegation APDM

새로 만들어진 위임 역할은 관리자와 위임자에 의한 폐지가 가능하다. 이에 따라 역할계층에서의 R1과 R2가 추가로 만들어지는 것은 다음 그림 7에서 보는 것과 같다.

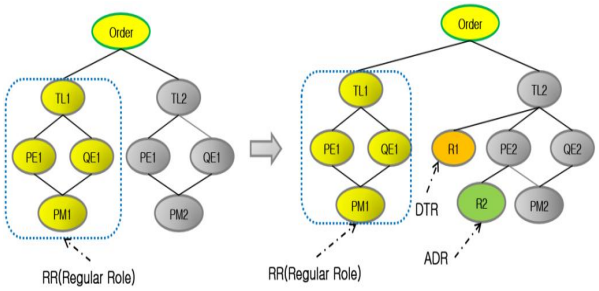

Fig. 7. The transition between the role hierarch

다음 그림 8은 제안하는 위임 모델이다. 제안하는 기법은 위임 역할에 속성을 추가하여 위임자에게 역할과 권한을 사상하고 위임된 역할은 관리자와 위임자에 의해 위임과 폐지를 관리하게 된다. 또한, 속성에 의해 위임자나 관리자가 관여하지 않아도 속성이 설정한 값을 벗어나면 위임은 자동으로 회수 또는 폐지되며, 위임받은 역할을 악용하거나 남용할 경우 위임자가 원하는 시점에 회수할 수 있다.

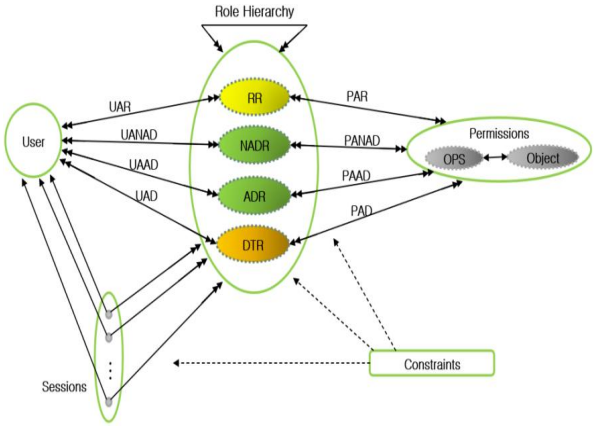

Fig. 8. The Proposed the APDM

3. Formal Definition of APDM

U:사용자 집합 S:세션 집합 R:역할 P: 권한 집합
DBR:위임할 수 있는 역할 RR:위임할 수 없는 역할
ADR:속성 값 중심의 단순위임 역할
NADR:속성 값의 다단계위임 역할
DTR:위임 역할 UAR: 사용자에 대한 역할 사상 관계
UANAD:사용자에 대한 다단계 속성 위임 사상 관계
UAAD:사용자에 대한 단순 속성 단순위임 사상 관계
UAD:사용자에 대한 위임 사상 관계
PA:권한에 대한 사상 관계
PAD:권한에 대한 위임 사상 관계
PAR:권한에 대한 역할 사상 관계
PANA:다단계 속성 권한에 대한 역할 사상 관계
PAAD:단순 속성 권한에 대한 역할 사상 관계

[정의4] 본 구성원의 역할은 RR, DTR, DBR로 구별되고, 위임 역할은 NADR, ADR로 구성되며 다대다의 관계이다.
-UAR⊆U×RR,UANAD⊆ U×NADR,UAAD ⊆ U×ADR
-DBR = NADR∨ADR,R=RR∨DBR∨DTR

[정의5] 위임 가능한 역할들과 위임 불가능한 역할 RR의 중복성이 없음을 나타낸다.
-RR∧NADR=Ø,RR∧ADR=Ø,DBR∧RR=Ø,DTR∧RR=Ø, DTR∧DBR = Ø

[정의6] 다단계위임이 발생할 때 UAD와 UAB로 위임 가능한 권한의 집합과 사용자 간 사상하는 함수 관계이다.
Permission*(u)={p:P| ∃r∈DTR,(r,p)∈PAD∧(u,r)∈UAD}∨{p:P| ∃r∈DBR,(r,p)∈PAB∧(u,r)∈UAB}

[정의7] NADR은 다단계 속성 위임을 말하고, ADR은 단순 속성 위임들의 집합이다.
-NADR⊆ DBR × A(Ab,Ae),A⊆TD -다중위임 가능
-ADR⊆ DBR × A(Ab,Ae),A⊆TD -단순위임만 가능
-(NADR ∨ ADR) ∧ DTR =Ø, NADR ∧ ADR =Ø

[정의8] 각 권한 사상은 위임된 역할 선정된 권한 p를 전부 또는 일부 포함한다.
-PAR⊆P×RR,PANAD⊆P×NADRn,
PAAD⊆ P×ADRn, PA=PAR∨PNAAD∨PAAD∨PAD

[정의9] 고정 역할을 사상 받는 구성원들은 위임 가능한 역할을 사상하고, 다른 이에게 자신의 고정된 역할을 사상할 수 없다.
-∀rr∈RR, ∃u:U,nadr:NADR,adr:ADR ▪ (u,rr)∈URA ∧rr=own_na(nadr)∧nadr=own_a(adr)⇒user_r(rr)=user_na(nadr)∧ user_na(nadr)=user_ac(adr)

[정의10] 위임 역할 집합을 다단계 속성 역할에 사상한다.
-own_d(r): NADR→2DTR and
(nadr1,nadr2∈NADR,dtr∈DTR) ▪ (nadr1≠nadr2)∧ (dtr∈own_d(nadr1)∧dtr∈own_d(nadr2))

이와같이 [정의5]는 상속 관계를 정의하여 역할의 중복을 방지한다. [정의7]은 [정의1], [정의2]와 [정의3]에 의해 속성을 활용하여 위임하게 된다.

4. The Select Permission Delegation

선정된 권한에 따른 접근 권한은 실제적인 단위가 역할이 아닌 업무다. 하지만 권한 관리를 업무 단위로 하는 것은 불가능하다. 그리고, 여러 특성이 있는 업무들이 있고, 그 특성에 따른 관리도 필요하다. 하지만, RBAC에서는 이를 지원하지 못한다.

그렇기에 제안된 모델에서 역할은 업무 집합들을 의미한다. 역할과 관련된 업무의 세분화를 통해 권한 일부분을 위임할 수 있게 한다. 그림 9에서 PL의 작업 p1(분석), p2(설계), p4(테스트)로 나눌 수 있고, 관리자로부터 권한을 제공받은 U1은 p4에 권한만을 다른 이에게 위임하고자 한다고 가정한다. 변경된 의미의 역할에서는 PL을 업무 단위로 세분화했기 원하는 위임 역할을 만들어 자신의 권한 중 업무 r4만을 선택해 위임할 수 있다.

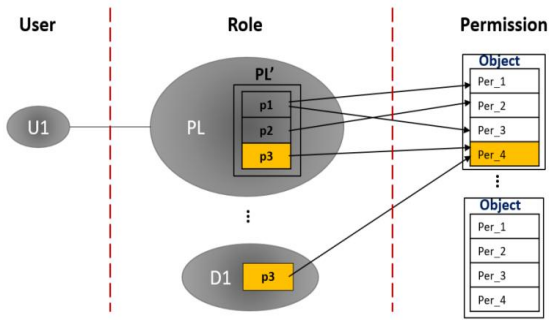

Fig. 9. Changed the Meaning of Delegation

5. Delegation in management

제안 모델의 상위 계층에서는 위임할 객체에 대해 권한을 제공받은 역할과 기존 역할이 갖는 한정 조건을 상속하여 한정함으로써 시스템상에서 위임 대상에 대한 한정적 선택을 가능하게 한다. 하지만, 기준을 만족하는 객체 사이에서 위임이 발생 하나더라도, 관리 허가 같은 경우 구성의 하위 역할에서의 위임을 해서는 안 될 권한위임이 일어날 수 있다. 그리고, 위임받는 다른 이가 그들의 업무를 적절하게 이행하지 않을 위험이 있다. 그래서, 역할의 이행이 위임자에 의해 관리돼야 하며, 동일 역할계층에서 일어난 위임도 위임이 적합한지에 대한 같은 범위 대한 판단을 해야한다.

위임 역할에서 ADR은 단순 속성 위임으로 생성된 역할이기에 다른 역할과 계층 관계를 성립하지 않는다. 그래서, 그런 역할을 관리할 만한 역할이 존재하지 않고, 한정 받은 속성값을 소비하면 자동으로 권한이 폐지된다. 그리고, 다른 역할과의 계층 관계를 성립하지 않기에 이를 보완하기 위해 위임 역할에 대한 관리를 관리자뿐만 아니라 위임자도 같이 위임 역할을 만든 역할이 위치한 역할계층에서 상위 역할이 가지는 관리, 권한 정보를 상속받는다. 따라서 새로 생성된 ADR 단순 속성 위임 역할에 대한 관리는 기존 역할에 대해 관리를 하고 역할계층 상의 상위 역할이 위임 역할에 대해 관리 감독한다. NADR 다단계 속성 위임 역할을 사상한 경우는 정적 사상이 일어난다. 권한을 소유한 역할에 위임을 촉진하여 해당 허가를 실행하려면, 세션 내에서 해당 역할을 촉진하기 위해 상위 관리 감독 역할에 의해서 정적 역할 대 역할 사상 후에서야 가능하다. 이것 없이 위임 역할에 대한 사상이 일어나면 사상된 역할을 세션에서 촉진 시키지 못해 작업할 수 없다.

6. An analysis of Delegation

위임 유형에서 역할기반 초기 모델인 RBAC모델은 위임을 고려하지 않았고, 반면에 RBDM0와 PBDM0은 단순위임을 지원하여 언제든 폐지가 가능한 위임을 고려하였다. 또한, PBDM은 단순위임과 다단계 위임을 동시에 지원한다. APDM은 NADR과 ADR로 인해 전자는 다단계위임을 후자는 단순위임을 지원함을 보였다.

부분 권한위임은 역할에 사상된 권한을 일부만 위임하는 것으로 RBAC모델은 위임보다는 상속의 개념이고, RBDM은 역할기반으로 위임해 자신의 권한 전체를 위임하는 형태를 나타내고 있으며, PBDM은 RR을 만들어 사용자에게 위임하는 방법으로 권한을 역할 부분집합에 속하게 하였고, APDM은 권한을 세분화하여 자신 권한에 속한 업무를 새로 생성된 역할에 사상하여 위임하였다. 이는 권한을 부분집합으로 나누어 권한 집합 내의 위임이기도 하다.

위임 역할 회수는 위임 역할이 새로 만든 역할에 대한 권한을 유지함으로써 새로 만든 역할의 남용이 나타나는 경우나 새로 만들어진 역할에 대한 역할 부정행위가 나타나는 등 위임 역할이 원하지 않는 경우, 언제든 폐지가 가능한 경우를 나티낸다. 새로 만든 역할에 위임 역할의 권한이 복제되어 위임되는 경우와 위임 역할의 권한도 함께 만들어진 역할에 위임되는 경우라 할 수 있다. 기존의 모델들은 모두 보안 관리자나 관리자가 역할을 폐지 또는 삭제하는 방식이고, APDM은 위임 역할이 새로 만든 역할의 부정행위를 방지하기 위해 역할을 감독하게 되고 동시에 관리자나 위임자 역할이 없이도 속성에 의해 자동으로 폐지되는 방식이다.

위임 관점에서 위임하는 대상이 사용자 입장에서의 위임인지 역할 입장에서의 위임인지를 나타내며, 기존의 모델들은 사용자 입장에서의 위임을 했으며, 실제 기업환경에서는 사용자-사용자 위임은 대부분 이루어지지 않고 있으며 역할 대 역할의 위임이 이뤄지고 있다. 다른 모델과 다르게 APDM은 역할 입장에서 역할-역할 위임방식이다.

IV. Conclusions

본 논문에서는 속성을 첨가한 권한의 위임에 중점을 두어 관리자는 모든 권한과 역할에 대해 관리할 수 있는 관리자 역할을 가지며, 역할들은 자신에 속한 권한들을 다른 이에게 위임을 하고자 하는 경우 관리자의 역할로 위임자가 위임하려는 역할을 복사하여 위임하게 된다. 또한, 위임에 대한 철회에서는 위임자와 역할 관리자가 함께 폐지 권한을 가지게 되고, 속성을 사용하여 적용된 속성에서 권한을 촉진하게 되며 역할 관리자나 위임자가 관여하지 않아도 자동으로 철회되는 위임 기법을 제안하였다.

향후 연구과제로는 위임을 속성에 대한 의미에서 접근하여 확장된 접근제어 모델을 연구할 것이며, 위임 기법이 적용된 다른 모델들과 비교하고자 한다.

REFERENCES

[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

[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

[5] Sandhu, R. (2005): Role Usage and Activation Hierarchies, http://www.list.gmu.edu/it862/it862s05/Role_Activation_Hierarchies.ppt. Accessed 16th February 2007.

[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, pp404-441, August, 2003,

[7] Crampton, J. and Khambhammettu, H. (2006): Delegation in Role-Based Access Control. Proc. 11th European Symposium On Research In Computer Security (ESORICS 2006), Hamburg, Germany,

[8] A Ali, U Habiba, MA Shibli "Taxonomy of delegation model", 12th international conference on information technology-new generations, IEEE, pp.218-223, 13-15 April 2015

[9] 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

[10] Jun Zheng, Yuan Tan, Qikun Zhang, Xin Sun, Yichun Chen, Applied Informatics and Communication, vol. 227, pp. 526, 2011.

[11] Tahmina Ahmed, Ravi Sandhu "Classifying and Comparing Attribute-Based and Relationship-Based Access Control" Conference: the Seventh ACM, March 2017

[12] X. Jin, R. Krishnan, and R. Sandhu. A unied"attribute-based access control model covering DAC,MAC and RBAC." In IFIP Annual Conference on Data and Applications Security and Privacy, Springer, p.41-55, 2012.

[13] 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,November14-16,2004

[14] Jiwan Ninglekhu, Ram Krishnan "AARBAC: Attribute-Based Administration of Role-Based Access Control", 2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC), IEEE, 15-17 Oct. 2017

[15] Koji Hasebe,Mitsuhiro Mabuchi,Akira Matsushita "Capability-based delegation model in RBAC", Proceedings of the 15th ACM symposium on Access control models and technologies, pp 109-118, June 2010

[16] D. R. Kuhn, E. J. Coyne and T. R. Weil, "Adding attributes to role-based access control", Computer, vol. 6, (2010), pp. 79-81.

[17] Bernhard J. Berger,Christian Maeder,Rodrigue Wete Nguempnang, Karsten Sohr,Carlos Rubio-Medrano(Less) "Towards Effective Verification of Multi-Model Access Control Properties"Proceedings of the 24th ACM Symposium on Access Control Models and Technologies, pp 149-160, May 2019

Authors

Si-Myeong Kim received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Dongguk University, Secoul, Korea, in 2003, 2006 and 2011, respectively. Dr. Kim joined the faculty of the Department of Computer Science at Dongguk University, Seoul, Korea, in 2014. He is currently a Adjunct Professor in the Department of Computer Science and Engineering, Dongguk University and CEO of YK ENT co.,Ltd. He is interested in access control and delegation and Computing security and Artificial intelligence security.

Sang-Hoon Han received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Dongguk University, Korea, in 1990, 1995 and 2002, respectively. Dr. Han joined the faculty of the Department of Computer Information Security at Korea National University of Welfare, Pyeongtaek, Korea, in 2003. He is currently a Professor in the Department of Computer Information Security, Korea National University of Welfare. He is interested in Information Security, Internet of Thing(IoT) and Computer Vision, and multimedia computing.