효과적인 역할계층 관리를 위한 기본 무결성 규칙

오 세 종†

요 약

ARBAC(adminstrative role-based access control)은 다수의 보안 관리자에 의한 분산 권한관리를 위한 대표적인 보안 모델이다. 각각의 보안 관리자는 역할계층 내에서 자신의 권한 관리 영역을 지정 받는다. ARBAC 모델의 문제중의 하나는 역할계층에 대한 합법적인 변경행위가 불법적 정보 흐름과 같은 원하지 않는 결과를 가져올 수 있다는 점이다. 이를 방지하기 위해 ARBAC 모델의 일부인 RRA97 모델에서는 역할계층의 기하학적 구조에 기초한 복잡한 제약조건을 제시하고 있다. 본 논문에서는 집합론에 기초한 단일 무결성 규칙을 제안한다. 이 규칙은 단순하고 직관적이며, RRA97 모델의 모든 제약조건을 대체할 수 있다.

키워드 : 정보보호, 접근제어, 역할, 역할계층

Master Integrity Principle for Effective Management of Role Hierarchy

Se-Jong Oh†

ABSTRACT

Administrative Role-Based Access Control(ARBAC) is a typical model for decentralized authority management by plural security administrators. They have their work range on the role hierarchy. A problem is that legal modification of role hierarch may induce unexpected side effect. Role-Role Assignment 97(RRA97) model introduced some complex integrity principles to prevent the unexpected side effect based on geometric approach. We introduce simple and new one integrity principle based on simple set theory. It is simple and intuitive. It can substitute for all integrity principles of RRA97 model.

Key Words : Security, Access Control, Role Hierarchy, ARBAC

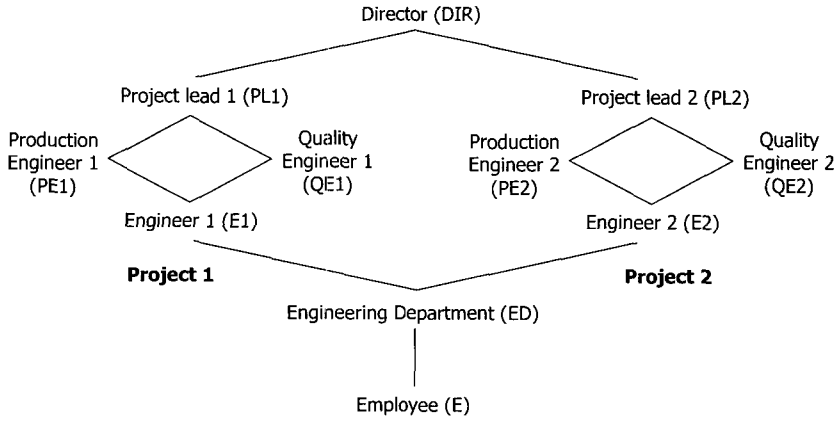
1. 서 론

역할기반 접근제어(RBAC ; role-based access control) 모델[1-4]은 대규모 조직 또는 대규모 정보시스템에 적합한 접근제어 모델로 잘 알려져 있다. RBAC 모델의 핵심적인 아이디어는 사용자가 직접 조직의 정보에 접근하는 것을 막고, 사용자에게 담당 업무에 적합한 역할(role)을 부여하며, 업무를 위해 필요한 최소한의 권한을 역할에 부여하자는 것이다. 사용자는 주어진 역할을 통해서만 권한을 행사할 수 있다. 역할의 개념은 기업 및 조직의 환경에서 왔기 때문에 RBAC 모델은 기업 및 조직 환경에서 발생하는 접근제어를 모델링하는데 적합하다. RBAC 모델의 장점중의 하나는 역할계층(role hierarchy)[8-10]의 개념을 통하여 기업의 조직 구조를 모델링할 수 있다는 것이다. 역할계층에서 부모 역할(parent role)은 자식 역할(child role)이 갖는 모든 권한을 계승(inherit) 받는다. 이러한 역할계층의 개념은 사용자의 권한 관리를 보다 용이하게 만드는 요소이다.

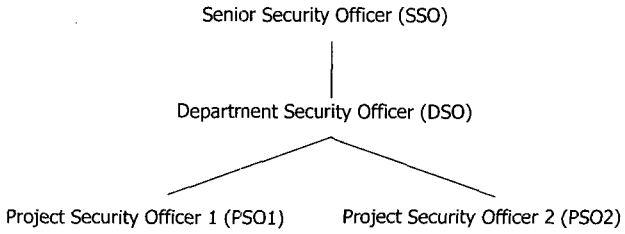
대규모 조직 또는 정보 시스템에서는 많은 수의 사용자와 많은 수의 정보 객체들이 존재하기 때문에 접근제어는 매우 어려운 일이 된다. 이러한 환경에서는 한명의 보안 관리자가 조직 전체의 접근제어를 관리할 수 없기 때문에 다수의 보안 관리자에 의한 분산 관리를 필요로 한다. ARBAC(administrative RBAC)모델[5, 6]은 이러한 접근제어의 분산 관리 환경을 지원하기 위해 제안되었다. RBAC의 관리 대상이 일반 사용자라면 ARBAC의 관리 대상은 보안 관리자들이다. ARBAC 모델은 (그림 1)과 같이 일반 사용자를 위한 일반역할계층[1]과 보안 관리자를 위한 관리역할계층을 포함한다. 분산 보안 관리를 위해 각 관리역할에는 관리 영역(administrative range)이 부여 되는데 (그림 2)와 같이 *can-modify* 테이블의 형태로 주어진다. 예를 들면 (그림 2)에서 관리역할 PSO1는 (그림 1)의 일반 역할계층에서 역할 E1~역할 PL1 사이의 관리 영역을 갖는다. 여기서 '관리'라 함은 보안 관리자가 관리 영역에 있는 역할들에 대해 권한을 부여하거나 회수할 수 있고, 관리 영역 내에서 새로운 역할을 추가하거나 삭제할 수 있음을 의미한다.

※ 이 연구는 2004학년도 단국대학교 대학연구비의 지원으로 연구되었음.
† 정 회 원 : 단국대학교 컴퓨터과학과 조교수
논문접수 : 2005년 6월 22일, 심사완료 : 2005년 10월 11일

1) 이후에 역할계층이라함은 일반역할계층을 의미한다.



(a) 일반 역할 계층

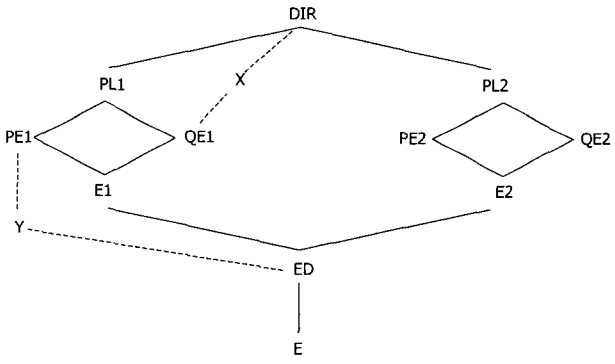


(b) 관리 역할 계층

(그림 1) 일반역할계층과 관리역할계층의 예

관리역할	권한영역
DSO	(ED,DIR)
PSO1	(E1,PL1)
PSO2	(E2,PL1)

(그림 2) *can-modify* 테이블의 예



(그림 3) 일반역할계층의 구조 변경에 따른 부작용의 예

ARBAC 모델에서 보안관리자가 자신의 관리 영역에 있는 역할들에 대해 권한을 부여하거나 회수하는데는 별 문제가 없다. 그러나 새로운 역할을 추가하거나 제거하는 경우 즉, 역할계층의 구조를 변경하는 경우는 예상치 못한 부작용(side effect)이 생길 수 있다. 예를 들면 일반역할계층이 (그림 3)과 같을 때 PSO1에 속한 보안 관리자가 역할 QE1을 역할 PE1의 부모 역할로 등록하게 되면(즉, PE1과 QE1 사이에 라인을 생성) PSO1의 관리 권한 밖에 있던 역할 Y의 권한이 역시 PSO1의 관리 권한 밖에 있는 역할 X로 계승되는 문제가 생긴다. ARBAC 모델에서는 이러한 부작용을 방지하기 위하여 다양한 제약사항을 두고 있다. 이에 대해서는 2장에서 자세히 설명한다.

일반역할계층 구조의 변경에 의한 부작용을 방지하기 위해 ARBAC 모델에서 채택하고 있는 방법은 역할계층의 기하학적 구조에 기반을 두고 있으며, 역할계층이 특정한 모양을 유지하도록 제약조건을 두고 있다. 이러한 접근 방식은 제약조건의 명확한 근거를 찾을 수 없을 뿐만 아니라 현재 존재하는 제약 조건을 피해가는 새로운 문제가 발견되면 그에 맞추어 추가적인 제약조건을 만들어야하는 맹점이 있다.

본 논문에서는 ARBAC 모델을 사용하는 분산 보안 관리 환경에서 ARBAC 모델의 문제점중의 하나인 역할계층 관리를 간단하면서도 효율적으로 수행할 수 있는 방법을 제안한다. 특별히 불법적인 권한변화를 초래하는 역할계층의 구조 변경을 탐지하여 막을 수 있는 방법을 제안한다. 본 논문에서 의미하는 역할계층 관리란 구체적으로 임의의 보안 관리자가

역할 계층상에 역할의 생성/삭제, 관계선의 생성/삭제를 시도할 때 이 행위의 적절성 여부를 판단하여 허용 및 금지하는 것을 말한다. 먼저 역할계층 구조의 변경에 의한 불법적인 권한변화의 판단 기준이 되는 단일 무결성 규칙을 제시하고, 간단한 집합론의 개념을 이용하여 불법적 변경이 발생 했는지를 판별하는 방법을 제안한다. 또한 제안한 무결성 규칙이 실제 상황에서 어떻게 불법적인 권한 변화를 판별할 수 있는지에 대한 예를 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 제시하며, ARBAC 모델에서 역할계층 구조의 변경에 따른 부작용을 방지하기 위하여 제시한 제약조건들과 그에 대한 문제점을 설명하고 기타 관련된 연구 결과들을 소개한다. 3장에서는 ARBAC의 문제를 극복하기 위한 기본 무결성 규칙과 부작용 판별 방법을 소개한다. 4장에서는 ARBAC의 접근방법과 본 논문에서 제안한 방법을 비교 분석하고, 5장에서 결론을 맺는다.

2. 관련 연구

2.1 ARBAC 모델에서의 역할계층 관리 및 문제점

ARBAC 모델은 URA97, PRA97, RRA97로 나뉘어지는데 역할계층의 변경에 대한 제약조건은 RRA97[7] 모델에서 다루고 있다. 역할계층의 변경은 네가지 연산—역할생성, 역할삭제, 관계선생성, 관계선삭제-에 의해서 이루어진다. 여기서 관계선이란 부모 역할과 자식 역할을 이어주는 라인을 말하며, 관계선을 통해 자식 역할의 권한이 부모 역할로 계승된다. 역할계층관리를 위한 몇 가지 정의와 역할계층을 변경시키는 네가지 연산에 대한 제약조건은 다음과 같다.

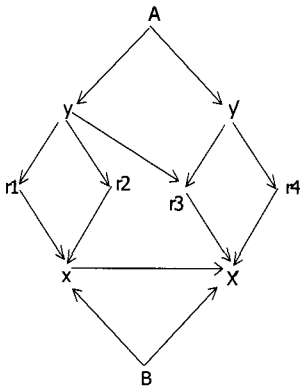
[정의 1] 역할의 영역(range)
(x,y) = {z∈role | x<z<y} where y>x. □

[정의 2] *can-modify* 테이블에서 기술된 사용된 역할의 영역을 권한영역(authority range)이라고 한다. □

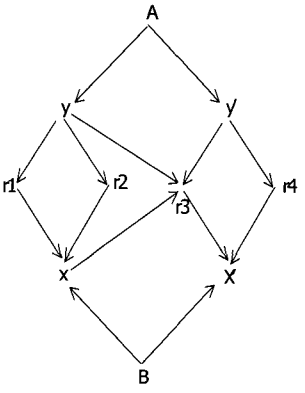
[정의 3] 역할영역 (x,y)가 다음 조건을 만족할 때 캡슐화되었다(encapsulated)라고 말한다.
-∀r1∈(x,y) ∧ ∀r2∉(x,y)일 때 r2>r1 ⇔ r2>y and r2<r1 ⇔ r2<x. □

[정의 4] 역할 r의 직속권한영역(immediate authority range) $AR_{immediate}(r)$는 다음과 같이 정의된다.
$AR_{immediate}(r)$ = (x,y) if r∈(x,y) ∧ r∉(x′,y′) for all (x′,y′) ⊂(x,y) □

[정의 5] 역할 영역 (x,y)는 다음과 같은 조건을 만족할 때 생성영역(create range)이라고 말한다.
-$AR_{immediate}(x)=AR_{immediate}(y)$ or x is end point of $AR_{immediate}(y)$ or y is an end point of $AR_{immediate}(x)$. □



(그림 4) 캡슐화영역 (x,y)



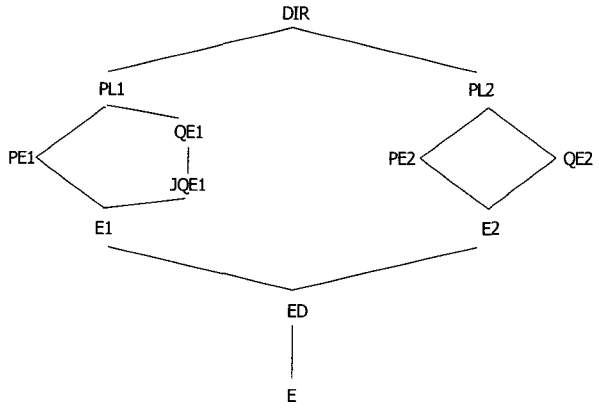
(그림 5) 비캡슐화영역 (x,y)

[IP1] (역할생성에 대한 제약조건) : 새로 생성하고자 하는 역할의 부모 역할과 자식역할이 생성영역일때만 그 사이에 새 역할을 생성할 수 있다.

[IP2] (역할삭제에 대한 제약조건) : *can-assign*, *can-revoke*[2], *can-modify* 테이블에서 참조하고 있는 역할은 삭제할 수 없다.

[IP3] (관계선생성에 대한 제약조건) : 역할 A와 역할 B 사이에 새로 생성되는 관계선 AB는 다음 조건을 만족하여야한다.
-$AR_{immediate}(A)$ = $AR_{immediate}(B)$ or
-권한영역 (x,y)가 (A = y ∧ B > x) ∨ (B = x ∧ A < y)일 때 관계선 AB의 생성후에 (x,y) 는 캡슐화 상태를 유지하여야 한다.

[IP4] (관계선삭제에 대한 제약조건) : 관계선 AB의 삭제가 역할 B의 권한을 변경한다면 관계선 AB는 삭제될 수 없다.



(그림 6) 변형된 역할계층

ARBAC에서 제시한 제약조건들은 쉽게 이해될 수 있으나 다음과 같은 단점이 있다.

2) *can-assign*는 보안관리자가 역할에 접근권한을 부여할 수 있는 영역을 나타내며 *can-revoke*는 반대로 접근권한을 회수할 수 있는 영역을 나타낸다.

- [IP1]은 (그림 3)과 같은 역할계층이 가진 잠재적 위험을 제거하는데 그 목적이 있다. ARBAC에서는 역할계층이 (그림 3)과 같이 되도록 허용하지 않는다. 그러나 (그림 3)까지는 문제가 없고, 관계선 PE1QE1이 생성될 때 비로소 문제가 발생한다. 따라서 [IP1]은 지나치게 제약적이다.
- (그림 6)과 같은 역할계층을 생각해 보자. (그림 2)의 *can-modify* 테이블이 (그림 6)을 위한 것이라 가정한다. 역할삭제의 제약조건인 [IP2]는 (그림 6)에서 PSO1가 JQE1을 삭제하는 것을 방지하지 못한다. 그러나 JQE1의 삭제는 QE1, PL1, DIR의 권한을 변경시키는데 PL1, DIR은 PSO1의 관리영역 밖에 있기 때문에 JQE1의 삭제는 불법이다.
- 앞에서 기술한 제약조건들은 캡슐화영역, 생성영역, 직속권한영역 등 다양한 개념들에 기초한다. 이에 대한 충분한 이해가 없이는 제약조건의 구현이 어렵다.
- 각각의 제약조건에 대한 이론적 근거가 모호하다. 이론적 근거 보다는 직관과 특정 사례들에 의하여 제약조건이 만들어졌다.
- 역할의 생성 및 삭제는 자연스럽게 관계선의 생성 및 삭제를 유발하기 때문에 [IP1]~[IP5]는 통합적으로 고려되어야 한다.

지금까지 살펴본 바와 같이 ARBAC 모델에서 제시하는 제약조건들은 실제 활용에 있어서 효율적이지 못하다. 3장에는 본 논문에서 제안하는 새로운 접근법을 소개한다.

2.2 ARBAC에서의 역할계층 관리의 문제점을 해결하기 위한 연구

역할계층에 관해서는 많은 연구가 있으나 대부분 접근제어 모델의 관점에서 연구가 이루어졌고 다수의 보안관리자를 위한 보안 관리 모델의 측면에서 이루어진 연구는 많지 않다. 접근제어 모델이 아닌 보안 관리 모델의 측면에서 효율적인 역할계층 관리가 본 논문의 주제이므로 이에 관련된 최근의 연구 결과들만을 소개하기로 한다.

Wedde[14]의 방법에서는 ARBAC 을 기초로 하고 있으면서 보안 관리자들에 대한 관리 영역의 할당 및 보안 요구사항을 술어 논리(predicate logic)유사한 언어에 의해 기술하고 역할계층의 변경시 이미 정의된 보안 요구사항을 위배하는지의 여부를 조사하여 변경의 허용 여부를 결정한다. 예를 들면 역할계층상에서 관계선을 삭제하려면 미리 정의된 'feature predicate'와 충돌을 일으켜서는 안된다. Wedde의 접근 방법은 기본적으로 ARBAC의 개념을 이용하고 있지만 보안정책의 표현에 있어서 언어적 방법을 주로 사용함으로써 매우 복잡하고 표현된 보안 정책들 사이의 무결성 검증 문제를 안고 있다.

OASIS 모델[11] 에서는 RBAC 모델에서의 역할계층이 현실세계에서는 유용하지 않기 때문에 역할계층을사용하지 않는다. OASIS 에서는 역할과 역할의 관계가 동적이며 역할활성화 규칙(role activation rule)에 의해 역할을 통한 접근을 제어하고 있다. OASIS는 접근제어에 대한 관리 모델이기 보다는 새로운 접근제어 모델이고 [14]와 같이 역할 및 규칙에 기초한 접근제어를 채택하고 있기 때문에 설정한 규칙에 대한 무결성을 설계자가 책임져야 하는 난점을 안고 있다.

SARBAC 모델[9, 12]을 통해 Crampton은 ARBAC 모델이 여러 면에서 안전성이 부족함을 지적하였고 주된 이유가 ARBAC의 캡슐화 영역(encapsulated range)의 개념에 있음을 밝혔다. Crampton은 이를 극복하기 위하여 관리 영역(administrative scope)을 제시 하였다. 관리 영역은 역할계층상에서 개별 보안관리자가 권한 관리를 할 수 있는 역할들의 영역을 의미하며 ARBAC에서는 권한범위(authority range)에 의해 정의되던 부분이다. ARBAC의 권한 범위는 정적이며 단순한 대신 제약조건이 많고, Crampton의 관리역역은 동적이며 복잡한 대신 제약조건이 적은 특징이 있다.

Koch[13]는 역할계층상에서 각 보안 관리자의 관리 영역을 그래프 이론을 이용하여 표현하고 관리하는 방법을 제안하였다. Koch는 제안된 방법을 SARBAC 모델에 적용한 새로운 모델을 제안하였는데 그래프 변환 규칙(graph transformation rule)에 의해 관리 연산을 표현할 수 있고 제약조건의 표현 및 검증이 용이한 장점이 있음을 보였다.

이번 절에서 제시한 선행 연구들은 ARBAC의 단점을 극복할 수 있는 대안으로서 제시되었다. 그러나 보안 정책의 표현에 있어서 규칙이나 언어를 사용함으로써 유연성이 높아진 대신 표현된 보안 정책에 대한 무결성 검증의 문제를 안고 있다. 또한 개발자의 입장에서 시스템으로 구현하기 어려운 복잡성을 안고 있다. 따라서 단순하면서도 효율적인 역할 계층 관리 방법을 필요로 한다.

3. 권한 집합에 기초한 새로운 역할계층 관리방법

대규모 조직에서 ARBAC 모델에 의하여 권한을 분산 관리하는 것은 쉬운 일이 아니다. 그 이유는 각 보안 관리자의 권한영역이 역할계층 위에 정의가 되어 있는데, 보안 관리자는 역할계층을 변경할 권한을 일부 가지고 있기 때문이다. 이렇게 역할계층을 변경하는 것은 불법적인 권한의 변화를 초래할 위험성을 안고 있다. 따라서 ARBAC의 과제는 보안관리자가 주어진 권한 영역 내에서 자유롭게 권한을 행사하게 하면서 동시에 불법적 권한 변화를 초래하는 역할계층의 변경을 효과적으로 방지하는 것이다. 이러한 목표를 달성하기 위해 ARBAC에서 제시하는 제약조건은 비효율적이기 때문에 새로운 역할계층 관리 방법이 필요하다. 본 논문에서는 역할계층의 '불법적 변경'이 무엇인지를 명확히 정의하여 새로운 대안을 찾고자 하였다. 본 논문에서는 '불법적 변경'을 다음과 같이 정의한다.

[정의 6] 역할계층의 불법적 변경

보안관리자 A가 자신의 관리영역 내에 있는 역할계층을 변경함으로써 관리역할계층상에서 상위에 있는 모든 보안관리자들의 관리영역에 있는 역할들 중 어느 하나에라도 권한의 변동에 생기게 한다면 보안관리자 A의 변경행위는 불법

이다.

예를 들면 (그림 3)의 역할계층과 (그림 2)의 *can-modify* 테이블에서 PSO1이 PE1과 QE1 사이에 관계선을 생성한다면 PSO1의 상위 관리자인 DSO의 관리 영역에 있는 역할 X의 권한에 변화가 발생한다. 따라서 PSO1의 변경행위는 불법이다. [정의 6]은 ARBAC 모델의 제약조건인 [IP1]~[IP4]에 내포되어 있던 내용을 명시적으로 표현한 것이다. 본 논문에서는 [정의 6]을 기본 무결성 규칙(master integrity principle)이라고 칭한다. 그 이유는 역할계층의 변경시 불법성 여부를 판단하는 기준이 될 수 있기 때문이다. 3.1절에서 기본 무결성 규칙을 보다 정형적으로 정의하도록 한다.

3.1 기본 무결성 규칙의 정의

기본 무결성 규칙의 정의에 앞서 필요한 보조 정의를 먼저 기술한다.

[정의 7] DAR(r) : 역할 r에 직접 할당된 접근권한(direct access right)의 집합

IAR(r) : 역할 r의 자식 혹은 자손 역할로부터 계승 받은 접근권한 (indirect access right)의 집합

TAR(r) : 역할 r이 가지고 있는 모든 접근권한(total access right)의 집합 TAR(r) = DAR(r) ∪ IAR(r)이다. □

(그림 1)에서 TAR(QE1) = DAR(QE1) ∪ IAR(QE1) = DAR(QE1) ∪TAR(E1) = DAR(QE1) ∪ DAR(E1) ∪ IAR(E1) = .. = DAR(QE1) ∪ DAR(E1) ∪ DAR(ED) ∪ DAR(E)이다.

[정의 8] 다음 조건을 만족할 때 역할 r2를 역할 r1의 조부모(senior parent)라고 하며 SP(r1)으로 표기한다.

-r1 ∈ (a,b) ∧ {r2,r3} ∈ (c,d) ∧ (a,b) ⊂ (c,d) ⇒ r1의 부모 또는 조상이면서 r2의 자식인 역할 r3는 존재하지 않는다. □

(그림 3)에서 SP(QE1) = {PL1, X}, SP(PE1) = {PL1}.

[정의 9] 권한영역(a,b)에 대한 조부모 역할은 SAR(a,b)로 표기하며 다음과 같은 조건을 만족하는 역할 r의 집합이다.

-r ∈ (c,d)

-(a,b) ⊂ (c,d)

-∃r2 : r2 ∈ (a,b) ∧ (r = SP(r2)) □

(그림 3)에서 SAR(E1,PL1) = {PL1, X}, SAR(E2,PL2) = {PL2}이다.

[정의 10] 보안 관리자가 역할계층을 변경하기 직전의 역할계층 구조를 BRH로 표기하고 역할계층을 변경한 이후 예상되는 구조를 ARH로 표기한다. □

[정의 11] 함수 *isValid()*는 BRH와 ARH를 비교함으로써 불법적 권한 변화가 있는지를 판단하는 기능을 수행하며 수행 알고리즘은 다음과 같다.

```
boolean isValid((x,y), BRH, ARH) {
// (x,y)는 변경이 발생한 권한 영역
  while(SAR(x,y) is not empty) {
    select r from SAR(x,y);
    a = TAR(r) of BRH ;
    b = TAR(r) of ARH ;
    if ( a ≠ b )   // 변경으로 인해 TAR(r)의 변화 발생
      return false ;
    remove r from SAR(x,y);
  }
  return true ;
}
```

함수 *isValid()*에서 역할계층상에 있는 모든 역할 r에 대해 DAR(r) ≠∅ 으로 처리 한다. 그 이유는 역할 r에 직접 할당된 역할이 없는 경우 *isValid()*가 정상적으로 실행되지 못하는 것을 방지하기 위함이다.

[정의 12] 기본 무결성 규칙 : 권한영역(a,b)에 대한 변경은 *isValid*((a,b), BRH, ARH)가 true일 때 합법적이다. □

기본 무결성 규칙은 보안 관리자가 역할계층을 변경할 때 합법성 여부를 판단할 수 있는 간단하고도 단일한 원칙이다. 다음절에서는 기본 무결성 규칙이 어떻게 역할계층 관리에 적용될 수 있는지 실제 사례로 설명한다. 앞에서 언급한 바와 같이 역할계층 관리란 구체적으로 임의의 보안 관리자가 역할 계층상에 역할의 생성/삭제, 관계선의 생성/삭제를 시도할 때 이 행위의 적절성 여부를 판단하여 허용 및 금지하는 것을 말한다.

3.2 기본 무결성 규칙의 적용

기본 무결성 규칙은 기본적으로 ARBAC의 역할의 생성/삭제, 관계선의 생성/삭제에 대한 모든 제약조건 [IP1]~[IP4]를 대치할 수 있다. 즉, [IP1]~[IP4]이 규정하고 있는 불법적 변경은 기본 무결성 규칙에 의해서도 불법으로 판정된다. 이번 절에서는 기본 무결성 규칙이 어떻게 불법적인 권한 변화를 판별할 수 있는지에 대한 예를 제시한다.

3.2.1 두 역할 사이에 관계선의 생성

앞에서 살펴본 바와 같이 (그림 3)의 역할계층상에서 PSO1가 PE1과 QE1 사이에 관계선을 생성(QE1이 PE1의 부모)하면 이것은 불법적 변경이다. 본 논문에서 제시한 기본 무결성

규칙에 따르면 역할 X의 전체 접근권한 TAR(X)에 변동이 생기면 불법적인 변경이다. 다음은 TAR(X)의 변동 여부를 확인하는 과정이다.

a = TAR(X) of BRH // 관계선의 생성전 역할 X의 접근권한의 집합
= DAR(X) ∪ IAR(X)
= DAR(X) ∪ TAR(QE1)
= ..
= DAR(X) ∪ DAR(QE1) ∪ DAR(E1) ∪ DAR(ED) ∪ DAR(E).

b = TAR(X) of ARH // 관계선의 생성후 역할 X의 접근권한의 집합
= DAR(X) ∪ IAR(X)
= DAR(X) ∪ TAR(QE1)
= DAR(X) ∪ DAR(QE1) ∪ TAR(E1) ∪ TAR(PE1)
= ..
= DAR(X) ∪ DAR(QE1) ∪ DAR(PE1) ∪ DAR(Y) ∪ DAR(E1) ∪ DAR(ED) ∪ DAR(E).
b - a = DAR(PE1) ∪ DAR(Y)
∴ b ≠ a
∴ PSO1가 PE1과 QE1 사이에 관계선을 생성하는 것은 불법으로 판정된다.

3.2.2 두 역할 사이의 관계선의 삭제

PSO1가 PL1과 QE1 사이의 관계선을 삭제하는 경우를 생각해 보자. 이 경우 PL1은 접근권한의 일부를 상실할 것을 예상할 수 있다. 기본 무결성 규칙에 따라 이경우 관계선의 삭제는 불법이다.

a = TAR(PL1) of BRH // 관계선의 삭제전 역할 PL1의 접근권한의 집합
= DAR(PL1) ∪ IAR(PL1)
= ..
= DAR(PL1) ∪ DAR(PE1) ∪ DAR(QE1) ∪ DAR(E1) ∪ DAR(Y) ∪ DAR(ED) ∪ DAR(E).

b = TAR(PL1) of ARH // 관계선의 삭제후 역할 PL1의 접근권한의 집합
= DAR(PL1) ∪ IAR(PL1)
= ..
= DAR(PL1) ∪ DAR(PE1) ∪ DAR(E1) ∪ DAR(Y) ∪ DAR(ED) ∪ DAR(E).
a - b = DAR(QE1)
∴ b ≠ a
∴ PSO1가 PL1과 QE1 사이의 관계선을 삭제하는 것은 불법으로 판정된다.

3.2.3 역할의 생성

역할계층상에서 역할을 생성하는 것 자체는 아무 의미가 없다. 역할이 의미 있기 위해서는 접근 권한이 부여되고, 역할계층상에 포함되어야 한다. 역할계층상에 포함되기 위해서는 기존에 존재하는 어떤 역할의 부모, 혹은 자식으로 관계선이 생성되는 것을 의미한다. 따라서 역할의 생성에 대한 타당성 여부는 3.2.1 의 두 역할사이의 관계선의 추가 문제로 환원하여 생각할 수 있다.

3.2.4 역할의 삭제

보안 관리자가 역할을 삭제하는 행위는 필연적으로 관계선의 삭제를 수반한다. (그림 3)에서 PSO1이 역할 QE1을 삭제하는 경우를 생각해 보자. 역할 QE1이 삭제되면 관계선 E1QE1, QE1PL1, QE1X가 삭제된다. 이 경우 PL1과 X는 접근권한의 일부를 상실할 것이다.

a = TAR(X) of BRH // 역할의 삭제전 역할 X의 접근권한의 집합
= DAR(X) ∪ DAR(QE1) ∪ DAR(E1) ∪ DAR(ED) ∪ DAR(E)
b = TAR(X) of ARH // 역할의 삭제후 역할 X의 접근권한의 집합
= DAR(X).

a - b = DAR(QE1) ∪ DAR(E1) ∪ DAR(ED) ∪ DAR(E)
∴ b ≠ a
∴ PE1이 QE1을 삭제하는 것은 불법으로 판정된다.

4. 제안한 방법과 다른 모델과의 비교

3.2절에서는 본 논문에서 제안한 기본 무결성 규칙이 ARBAC 모델의 제약조건 대신 사용될 수 있음을 보였다. 본 논문에서 제안한 방법은 ARBAC 모델에 비해 다음과 같은 장점이 있다.

- 본 논문에서 제안한 방법은 ARBAC 모델에 비해 단순하고 직관적이다. ARBAC 모델의 역할계층 관리 방법을 이해하려면 제시된 여러 개념들을 충분히 이해하여야 하지만 본 논문에서는 보안관리자라면 누구나 쉽게 납득할 수 있는 단일 개념을 가지고 역할계층 변경의 불법성 여부를 판단한다.
- 기본 무결성 규칙은 명확한 논리적 근거를 가지고 있다. ARBAC 모델의 제약 조건들은 왜 그러한 제약조건이 도출되었는지 논리적으로 설명하지 못한다. 따라서 제시된 제약조건들이 역할계층의 불법적 변경을 모두 차단할 수 있다는 확신을 할 수 없다. 본 논문에서는 [정의 6]의 불법적 변경에 대한 명확한 정의를 기초로 기본 무결성 규칙을 제안하였기 때문에 [정의 6]에 오류가 없는 한 기본 무결성 규칙은 불법적 변경을 판별하는데 명확한 근거가 된다.

- 기본 무결성 규칙은 하나의 기준을 가지고 역할계층의 불법성 여부를 판단한다. 그에 비해 ARBAC 모델은 역할의 생성/삭제, 관계선의 생성/삭제에 따라 서로 다른 규칙을 적용하기 때문에 복잡하다. 기본 무결성 규칙의 단순성은 분산 접근제어 시스템의 구현을 용이하게 한다.
- ARBAC 모델의 제약조건에 따르면 (그림 3)과 같은 역할계층 구조는 만들어질 수 없다. 그러나 기본 무결성 규칙에 의하면 (그림 3)은 문제가 없기 때문에 생성을 허용한다. 따라서 기본 무결성 규칙은 ARBAC 모델 보다 유연한 역할계층 관리를 제공한다.
- 기본 무결성 규칙은 역할계층을 변경하는 모든 작업에 적용하기 때문에 2장에서 제시한 IP2의 문제점을 발생시키지 않는다.

ARBAC 모델에서 역할계층 관리의 문제점을 해결하고자 한 다른 방법들과 본 논문에서 제안한 방법을 비교해 보면 우선 제안한 방법이 단순성과 구현성에 있어서 뛰어나다고 할 수 있다. 그 이유는 제안한 방법은 설계자가 보안 관리를 위한 정책을 규칙이나 언어에 의해 기술하는 방법이 아닌 단일 규칙을 사용하는 방법이기 때문에 특정 보안관자의 역할 생성/삭제, 관계선의 생성/삭제시 불법성여부를 시스템이 쉽게 판단할 수 있기 때문이다. 역할계층 관리에 규칙이나 언어를 이용하지 않는 SARBAC 모델의 경우는 해결하고자 했던 문제의 초점이 다르기 때문에 본 논문에서 제안한 방법과 상호 보완적으로 사용될 수 있다.

결론적으로 본 논문에서 제안한 방법이 ARBAC 및 기타 모델의 방법보다 훨씬 단순하면서도 보다 유연한 역할계층 관리를 가능하게 한다.

5. 결 론

대규모 조직이나 정보 시스템에서는 다수의 보안 관리자에 의해 접근제어가 분산 관리되기 때문에 보안관리자들이 주어진 권한 내에서 보안관리 업무를 수행할 수 있도록 통제해야할 필요가 있다. 분산 보안관리의 어려운 점은 역할계층의 관리이다. 역할계층은 보안 관리자들의 권한영역을 규정할 수 있는 수단이면서 동시에 보안 관리자들의 관리 대상이기 때문이다. 본 논문에서는 기본 무결성 규칙을 통하여 단순한 방법으로 역할계층의 변경에 대한 합법성 여부를 판단할 수 있는 방법을 제안하였다. 제안한 방법은 단일한 규칙을 가지고 역할의 생성과 삭제, 관계선의 생성과 삭제의 모든 변경작업에 대해 합법성 여부를 판단할 수 있도록 하였다. 또한 기본 무결성 규칙은 간단한 집합론에 기반을 두고 있기 때문에 실제 시스템으로 구현하기도 용이하다.

역할에 부여된 접근권한을 집합으로 다루고자 하는 본 논문의 시도는 역할계층의 관리를 넘어서 역할 관리(role management)라는 보다 넓은 영역에도 적용할 수 있으며 현재 이에 대한 연구가 진행중에 있다.

참 고 문 헌

[1] R. Sandhu, "Rationale for the RBAC96 Family of Access Control Models", Proc. of the First ACM Workshop on Role-Based Access Control, 1995.

[2] D. Ferraio, J. Cugini, R. Kuhn, "Role-based Access Control (RBAC) : Features and motivations", Proc. of the 11th Annual Computer Security Application Conference, 1995.

[3] R. Sandhu, E. Coyne, H. Feinstein, C. Youman, "Role-Based Access Control Models", IEEE Computer, Vol.29, No.2, 1996.

[4] S.I. Gavrila, J.F. Barkley, "Formal Specification for Role Based Access Control User/Role and Role/Role Relationship Management", Proc. of the 3rd ACM workshop on Role-Based Access Control, 1998.

[5] R. Sandhu, Bhamidipati, Q. Munawer, "The ARBAC97 Model for Role-Based Administration of Roles", ACM Trans. on Information and Systems Security (TISSEC), Vol.2, 1999.

[6] R. Sandhu, Q. Munawer, "The ARBAC99 Model for Administration of Roles", Proc. of the Annual Computer Security Applications Conference, 1999.

[7] R. Sandhu, Q. Munawer, "The RRA97 Model for Role-Based Administration of Role Hierarchies", Proc. of the Annual Computer Security Applications Conference, 1998.

[8] J.D. Moffett, E.C. Lupu, "The Uses of Role Hierarchies in Access Control", Proc. of the 4th ACM Workshop on Role-Based Access Control, 1999.

[9] J. Crampton, "Administrative Scope and Role Hierarchies Operations", Proc. of the 7th ACM Symposium on Access Control Models and Technologies, 2002.

[10] J. Crampton, "On permissions, Inheritance and Hierarchies", Proc. of the 10th ACM Conference on Computer and Communication Security, 2003.

[11] W. Yao, K. Moody, J. Bacon, "A model of OASIS role-based access control and its support for active security", Proc. of the sixth ACM symposium on Access control models and technologies, 2001.

[12] J. Crampton, G. Loizou, "A foundation for role-based administrative models", ACM Transactions on Information and System Security (TISSEC), 2003.

[13] M. Koch, L. V. Mancini, F. Parisi-Presicce, "Administrative scope in the graph-based framework", Proc. of the ninth ACM symposium on Access control models and technologies, 2004.

[14] H. F. Wedde, M. Lischka, "Cooperative role-based administration", Proc. of the eighth ACM symposium on Access control models and technologies, 2003.

오 세 종

e-mail : sejongoh@dankook.ac.kr

1989년 서강대학교 컴퓨터학과(학사)

1991년 서강대학교 대학원 컴퓨터학과(공학석사)

2001년 서강대학교 대학원 컴퓨터학과(공학박사)

1991년~1997년 대우정보시스템 근무

2001년~2003년 미국 George Mason University Post Doc. 연구원

2003년~단국대학교 공학대학 컴퓨터과학과 조교수

관심 분야 : 정보시스템, 정보보호, 데이터베이스, 유비쿼터스 컴퓨팅