通信情報保護學會論文誌
第 8 卷 第 4 號, 1998. 12

# 역할 속성을 이용한 역할기반 접근통제 매커니즘

이 철 원**, 이 병 각**, 김 기 현**, 박 정 호**, 이 홍 섭**, 최 용 락*

# Role Based Access Control Mechanism Using Role Attributes

Cheol Won Lee**, Byung Kak Lee**, Ki Hyun Kim**, Chung Ho Park**,
Hong Sub Lee**, and Yong Rak Choi*

## 요 약

역할기반 접근통제는 현재 가장 많이 사용되는 임의적 접근통제 및 강제적 접근통제에 비하여 유연성을 제공하며 접근통제 기법이 정교하다. 따라서, 역할기반 접근통제는 금융 및 기업 등과 같은 상업적 분야 및 행정적인 분야에서 많이 사용될 것으로 예측된다.

본 논문에서는 기존에 제안된 역할기반 접근통제 메커니즘에 역할의 속성을 정의하여 새로운 형태의 역할기반 접근통제 메커니즘을 제안하였다. 제안된 역할기반 메커니즘은 기존의 역할기반 메커니즘이 실생활에 잘 적용되지 않는 단점을 극복하였으며, 특히 살라미 공격, 내부자의 부정행위 방지에 효과적으로 적용 가능하다. 또한 역할을 부여받는 사용자의 역할 중복 방지를 위하여 차이 연산자를 정의하였다.

## Abstract

Role based access control (RBAC) is more flexible and fine control than the discretionary and mandatory access controls that are currently popular used. Therefore, RBAC will be used in commercial and governmental sectors such as financial, enterprise, and etc.

In this paper, we add role attributes to the existing definition of RBAC mechanism and propose a new RBAC. The proposed RBAC mechanism shall be shown that it is suitable for real-world application. Especially, the proposed RBAC mechanism can be applied to effectively protect Salami attack and fraud by insiders and we also define difference operator to prevent the duplication of user's role.

* 대전대학교
** 한국정보보호센터

## I. 개 요

역할기반 접근통제(RBAC, Role-Based Access Control)의 개념은 1970년대 다중사용자와 다중응용을 위한 온라인 시스템에서 시작되어 현재 접근통제의 표준인 전통적인 강제적 접근통제(MAC, Mandatory Access Control) 및 임의적 접근통제(DAC, Discretionary Access Control)의 대안으로서 많은 관심을 집중시키고 있다. 역할기반 접근통제의 중요한 동기는 관리자가 수행하기 어려운 보안관리 과정을 능률적으로 처리하고 공공기관 및 기업에 특정한 보안정책을 명료하게 표현하고 시행하기 위함이었다[1][4]. 역할기반 접근통제는 관리자에게 누가, 언제, 어디에서, 어떤 행동을 수행할 수 있는지 규정할 수 있는 능력을 제공하여 준다.

기능적인 측면에서 역할기반 접근통제의 핵심 개념은 역할과 관련된 행동을 나타내는 연산(operation)과 역할(role)의 구성원으로 표현될 수 있는 사용자(user)이다. 사용자와 역할은 다대다(many-to-many) 관계를 가지고 있다. 예를 들어, 한 사용자는 하나 이상의 역할과 관련될 수 있고 하나의 역할은 한명 이상의 사용자를 가질 수 있다. 역할은 한 조직의 일에 따라 다양한 형태로 생성될 수 있다. 예를 들어, 은행의 경우 역할은 금전출납계원(Teller) 및 대부계원(Loan officer) 등을 포함할 수 있고 병원의 경우에는 의사, 간호사, 임상의 등의 역할을 포함할 수 있다. 역할과 관련된 연산은 역할의 구성원에게 특정 행동을 수행하도록 강요한다. 예를 들어, 병원의 경우 의사라는 역할은 진료, 조제, 수술 등의 연산을 포함하며 임상의학자는 연구를 위하여 익명의 임상정보를 수집하는 연산에만 국한된다.

역할기반 접근통제는 다음과 같은 장점을 가진다[1].

첫째, 관리자에게 편리한 관리 능력을 제공한다. 전통적인 접근통제 메커니즘의 경우 사용자의 접근권한 관리는 매우 성가신 작업이다. 그러나, 역할기반 접근통제의 경우 사용자의 자격과 책임에 따라 역할의 구성원으로 사용자를 지정하고 부여된 사용자의 업무에 따라서 사용자를 역할의 구성원에서 제외하고 새롭게 추가하는 것이 쉽게 이루어질 수 있다. 역할기반 접근통제에서는 연산은 사용자 개인별로 어떤 연산을 수행하도록 허가하는 것이 아니라 오로지 역할과 관계가 있으므로 조직의 기능 변화에 따라 역할과 관련한 연산의 삭제 및 추가 역시 자유롭게 이루어질 수 있다. 즉, 사용자 개인별로 접근권한을 설정하는 것이 아니라 사용자에게 부여된 임무를 기반으로 역할을 설정하고 이 역할에 허용된 연산을 허용함으로써 조직의 기능 변화에 따른 관리적 업무의 효율성을 꾀할 수 있다.

둘째, 접근을 통제하고자 하는 객체단위로 접근통제를 수행하는 기존의 방법(예, 접근통제리스트, 자격리스트 등)과는 달리 관리자는 역할, 역할계층(hierarchy), 관계(relationship), 제약(constraint)의 정립을 통하여 사용자의 행동을 정적 또는 동적으로 규제할 수 있으므로 시스템 관리자에게 객체단위가 아닌 추상적인 개념으로 접근을 통제할 수 있다. 따라서, 역할기반 접근통제를 업무를 수행하는 실제 환경에 자연스럽게 접목될 수 있다.

셋째, 역할기반 접근통제가 분산환경에서 사용되는 경우 역할기반 접근통제 관리자의 책임을 중앙과 국지 보호 영역으로 구분할 수 있다.

역할기반 접근통제는 은행 또는 병원과 같이 특정한 일에 대하여 특정한 일을 수행하는 분야에는 효과적인 접근통제 방법이다. 개인 정보의 보호, 금융 자산에 대한 권한 없는 접근의 방지, 무단 정보 사용의 방지 등 다양한 분야에서 역할기반 접근통제는 활용될 수 있다. 그러나, 역할기반 접근통제의 유용성에도 불구하고 역할기반 접근통제의 특징이 대규모 시스템에 적용되었을 경우 많은 역할과 역할 허가 사이의 복잡한 관계 설정

으로 인하여 실제 시스템을 완전히 통제하지는 못한다. 즉, 모든 접근통제 이슈들에 대한 해결책은 되지 못한다[4][5].

역할기반 접근통제를 실세계에 적용할 경우, 사용자에 역할을 할당하면 그 사용자는 할당된 역할에서 허가된 모든 연산을 수행할 수 있다. 미리 주어진 제약 조건들에 의해 사용자가 역할에 할당되고 역할에 할당된 연산을 수행하지만 권한 내에서 부정 조작을 하려고 마음먹으면 손쉽게 처리할 수 있다. 일반적인 예로서 은행의 경우를 보면 고객의 계정을 다루는 계정 책임자의 역할에 할당된 직원은 쉽게 계정 내역을 조작할 수 있다. 또한 대표적인 은행 공격의 예로서 살라미(Salami) 조작을 수행할 수 있다. 이것은 휴면계좌나 소수점 이하로 떨어지는 금액을 특정계좌로 모아 가로채는 수법으로 피해자가 손실이 매우 적기 때문에 감지하기가 어려우며 세심한 관리가 필요하다.

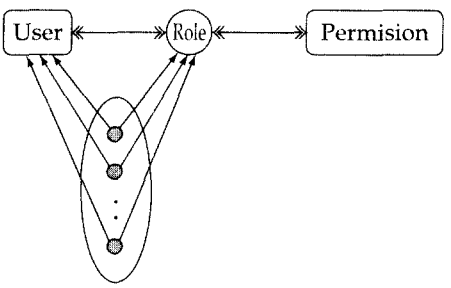
또한 역할계층상에서 상위에 존재하는 역할은 하위 역할의 연산을 상속받기 때문에 상위의 역할에게 많은 연산이 허용된다. 이러한 중복된 연산 허용은 부정조작 등의 위험성을 내포하고 있다.

본 논문에서는 이러한 문제점을 방지하기 위해 각 역할에 속성을 정의하였다. 정의된 속성은 시간, 지역, 트랜잭션의 횟수, 처리할 수 있는 돈의 액수 등이며 이러한 속성은 상호 결합하여 사용할 수 있도록 하였다. 이와 같이 정의된 역할 속성을 이용하여 이론적인 역할 기반 접근통제 모델을 실세계에서 구현 가능하도록 모델을 제안하였다. 또한 역할에 부여된 연산의 중복을 최대한 배제할 수 있도록 차이 연산자를 추가하였다.

본 논문의 구성은 다음과 같다. 제2장에서는 역할기반 접근통제 모델에 대하여 고찰하고 제3장에서는 역할속성을 이용한 역할기반 접근통제 모델을 제시하였으며 제4장에서는 제안된 역할기반 접근통제 모델을 실생활에 적용된 예를 제시하였으며 결론 및 향후 연구과제를 제5장에서 기술하였다.

## II. 역할기반 접근통제 모델

역할기반 접근통제는 최근들어 많은 연구가 수행되고 있으며 대표적인 역할기반 접근통제모델로는 [1], [2], [3] 등이 있다. 본 논문에서는 Ravi S. Sandhu 등이 제안한 역할기반 접근통제 모델에 대하여 소개하겠다[3].




(그림 1) 기본 역할기반 접근통제 모델

역할기반 접근통제 모델은 사용자(*U, User*), 역할(*R, Role*), 허가(*P, Permission*)의 세가지 엔티티들로 정의된다. 사용자는 컴퓨터의 네트워크, PC, 로봇과 같은 도구들도 포함할 수 있으나 일반적으로 사람이라고 정의한다. 역할은 역할의 구성원에게 수여된 책임과 권한에 관련된 의미를 가진 조직내의 직무 기능이나 직무 이름이다. 허가는 시스템내에 하나 이상의 객체(object)에 대한 접근 승인이다. 허가에서 객체는 컴퓨터 시스템 내의 자원뿐만 아니라 데이터도 포함된다.

사용자는 여러 역할들의 멤버가 될 수 있고 역할은 여러 사용자들을 가질 수 있다. 유사하게 역할은 많은 허가를 가질 수 있고 허가는 여러 역할들에 할당될 수 있다. 이러한 두 관계들을 나타내는 것이 사용자 할당(UA)과 허가 할당(PA)이며 다음과 같이 정의된다.

- $PA \subseteq P \times R$, 다대다 허가 대 역할 할당 관계(a many-to-many permission to role assignment relation)

- UA ⊆ U × R, 다대다 사용자 대 역할 할당 관계 (a many-to-many user to role assignment relation)

사용자는 워크스테이션에서 여러 윈도우를 여는 것처럼 동시에 여러 세션을 열 수 있으며 다음과 같이 정의된다.

- user : S → U, a function mapping each session $s_i$ to the single user user($s_i$) (constant for the session' s lifetime)
- roles : S → 2R a function mapping each session $s_i$ to a set of roles roles($s_i$)⊆{r | (*user*($s_i$), $r$)∈*UA* }(which can change with time) and session $s_i$ has the permissions $U_{r \in roles(s_i)} \{p \mid (p, r) \in PA\}$

여기서 세션(S, Session)은 한 사용자와 역할의 가능한 매핑을 나타낸다. 이 역할기반 접근통제 모델은 최소 권한(least privileges)의 원칙을 지원한다. 여러 역할의 구성원인 사용자는 그 세션에 수행되는 작업을 위해 이 역할들의 일부를 실시할 수 있다. 그러므로 역할의 구성원인 사용자는 정상상태에서 이 역할을 활성화하지 않다가 필요할때 활성화할 수 있다. [그림1]과 같은 역할기반 접근통제 모델은 세션이 살아있는 동안 역할들이 동적으로 활성화되고 비활성화되도록 허가한다. 세션의 개념은 전통적인 접근통제 개념에서 주체(subject)와 같다. 주체는 접근통제의 단위이고 사용자는 동시에 다른 허가를 활성화시키는 여러 주체들을 가질수 있다.

이와 같은 역할기반 접근통제 기본 모델은 역할계층(role hierarchy) 개념 및 제약(constraints)을 부가하여 진보된 형태의 모델로 발전시킬 수 있다. 역할 계층 및 제약에 관한 자세한 내용은 참고문헌 [2]에 기술되어 있다.

위와 같은 역할기반 접근통제 모델을 실세계에 반영하기 위한 많은 노력들이 수행되어 왔다. Luigi Giuri[4]는 and-role 및 or-role 연산자를 이용하여 보다 다양한 역할을 정의하였다. 그러나 실제 역할기반 접근통제를 시스템에 구현할 때 내부자의 부정행위 등을 방지하기 위해서는 역할뿐 아니라 역할에 부여된 속성이 필요하게 된다. 본 논문에서는 역할기반 접근통제를 실생활에 적용할 때 시스템의 구현에 필요한 역할 속성들을 정의하고 이를 이용하여 살라미 공격, 내부자 부정행위 등과 같은 공격에 능동적으로 대처할 수 있는 역할기반 접근통제 메커니즘을 제시한다.

## III. 역할 속성이 부가된 RBAC 모델

역할기반 접근통제 모델의 기본요소는 사용자, 역할, 허가(연산과 객체)이다. 이 논문에서는 여기에 속성이라는 새로운 요소를 부가한다. 이러한 요소들과 이들 사이의 관계는 아래와 같이 정의된다. 역할기반 접근통제에 대한 표현들은 여러 논문들에서 다르게 정의되었으나 기본 모델은 [6]에서 정의된 것을 따른다.

[역할기반 접근통제 모델에서 사용되는 요소]

- u : 사용자 (USER)
  - 사용자는 사람의 집합으로 시스템을 사용하는 신뢰성이 있는 사람 및 신뢰성이 없는 사람을 의미한다.
- x : 주체 (SUBJECT)
  - 각 사용자를 대신하여 역할내의 연산을 수행하는 시스템의 능동적인 실체(entity)를 의미한다.
- r,i,j : 역할 (ROLES)
  - 조직의 임무 또는 업무의 집합을 의미한다.
- op : 연산 (Operation)
  - 시스템의 객체에 허가되어진 접근모드의 집합 (eg. Read, Write, Create)

- o : 객체 (Object)
  - 권한없는 사용으로부터 보호되는 시스템의 수동적 실체의 집합 (eg. Password file, Account file)
- a : 속성 (Attributes)
  - 역할에 허가 권한을 부여하기 위해 새로이 정의한 속성들의 집합 (eg. Time, Area, Amount, NoofTransactions)
- p : 허가 (Permission) : Subset(Operation * Object * Attributes)
  - 객체에 대해 수행할 수 있는 연산과 이에 부가된 속성들의 조합(〈op, o, a〉)으로 기존 모델에 새로이 속성 요소가 추가된다.

[역할기반 접근통제 모델에서 사용되는 정의]

- authorizedroles : USER -> 2^ROLE
  - authorizedroles(u)은 사용자 u에 권한이 부여된 역할들의 집합 여기서, ^은 멱승을 의미한다.
- authorizedusers : ROLE -> 2^USER
  - authorizedusers(r)은 주어진 역할에 대해 권한이 부여된 사용자들의 집합
- authorizedperm : ROLE -> 2^Permission
  - authorizedperm(r)은 역할 r에 권한이 부여된 허가들의 집합으로 속성 조건이 부가되어 처리된다.
- activeuser : SUBJECT -> USER
  - activeuser(x)는 Subject x에 사용자의 연관성을 표현
- activerole : SUBJECT -> 2^ROLE
  - activerole(x)는 Subject x가 활성화되어 있는 역할들의 집합

[추가된 속성을 이용한 정의와 연산]

- permitop : ROLE * Operation * Object * Attributes -> Boolean
  - permitop(r,op,o,a)은 역할 r이 객체 o에 연산 op를 속성 a에 맞게 수행할 수 있으면 True의 결과값을 돌려준다.
    -〉 $\forall r \forall op \forall o \forall a$ permitop(r,op,o,a)
    $\Leftrightarrow \exists r$ (r∈active_role(x) ∧ p∈ authorized_ perm[r] ∧ <op,o,a > ∈p)
- PS : (Attributes -> 2^Attributes) -
  - PS(r, Attributes)는 역할 r에 허가 권한을 부여하기 위해 부가된 정의된 속성들의 멱집합(Power Set)을 의미한다.
    a∈PS(r, Attributes), Attributes = {attrib1, attrib2, ···, attribn}

[차이 연산]

역할 계층에서 상위 역할이 하위 역할로부터 허가들을 상속받게 된다. 이러한 상속받은 권한들을 일반적인 경우 사용하지 못하게 하고 상위 역할에 권한이 집중되는 것을 막기 위해 이 연산을 정의한다.

- differop : Permission * Permission -> Permission
  - differop(i,j) = authorizedperm(i) - authorizedperm(j) 여기서, role i는 role j보다 상위 역할임

역할기반 접근통제 모델에 속성이 추가되어도 참고문헌[6]에서 제시된 모델의 정적 의무분리 (SSD, Static Separation of Duties), 동적 의무분리 (DSD, Dynamic Separation of Duties), 역할 계층 상속(Role Hierarchiy Inheritance) 정의와 기능 (function)에는 변화가 생기지 않는다.

## IV. 역할기반접근통제 모델의 적용

본 장에서는 위에서 정의한 역할 속성이 부가

된 역할기반 접근통제 모델을 실제 은행시스템에 적용한 사례를 기술하였다. (그림 2)는 은행시스템에서의 역할기반 접근통제 모델에서 역할 계층의 예를 보여준다[7].

역할 **은행책임자(Bankrepresentative)**는 역할 **계정책임자(Accountrepresentative)**로부터 상속받는다. 즉, 은행책임자로서 할당된 사용자는 계정책임자 역할에서 허가받은 모든 작업을 수행할 수 있으므로 계정을 만들거나 삭제하는 것이 가능하다. **계정책임자**, **지점장(Branchmanager)**, **내부감사자(Auditor)**, 창구직원**(Teller)** 등은 모두 은행 고용 직원들이므로 해당하는 모든 역할은 **은행직원(Employee)** 역할로부터 상속받는다.

그림에서 사각형으로 표시된 **창구직원**과 **계정소유자(Accountholder)**는 각각 계정책임자 역할과 동적 의무 분리(Dynamic Separation of Duties) 관계가 된다. **계정책임자**로서 일하는 사람은 동시에 창구직원 또는 **계정소유자**의 역할을 할 수 없다. 은행의 고용직원인 **계정책임자**가 그 은행의 계정을 가질 수는 있지만 다른 사람의 계정을 처리하면서 동시에 자신의 계정을 처리할 수 없다. 또한 **계정책임자**가 **창구직원**의 역할을 맡을 수는 있지만 동시에 같이 처리할 수는 없다.

육각형으로 표시된 **내부감사자**는 **계정책임자**와 정적 의무 분리(Static Separation of Duties) 관계가 성립된다. 이것은 동적 의무 분리보다 더 강력한 관계를 표현한다. 만약 두 역할들이 동적 의무 분리 관계에 있을 경우는 한 개인에 대해서 두 역할의 권한이 부여될 수 있지만 동시에 수행될 수 없다는 것이고 정적 의무 분리 관계는 같은 사람에게 두 역할이 부여될 수 없다는 것이다. **내부감사자** 및 **계정책임자**의 역할은 매우 중요해 내부감사자 역할과 **계정책임자** 역할 사이에는 근본적으로 상호 배타가 필요하므로 같은 개인에게 동시에 권한이 부여될 수 없다.

이밖에 시스템 전반적인 관리를 위해 **시스템관리자**와 **보안관리자**의 두 역할을 정의한다. **시스템관리자**는 시스템의 물리적 자원들을 통제하고 관리하며 보안관리자는 각 역할들을 정의하고 사용자와 수행가능한 권한을 할당하는 등의 보안에 관련된 작업을 수행한다. 여기서 **보안관리자** 및 **시스템관리자**의 역할을 분리하여 각 역할에 맞는 최소 권한을 부여함으로써 시스템을 관리하는 관리자에 의한 보안사고를 예방할 수 있도록 하였다.

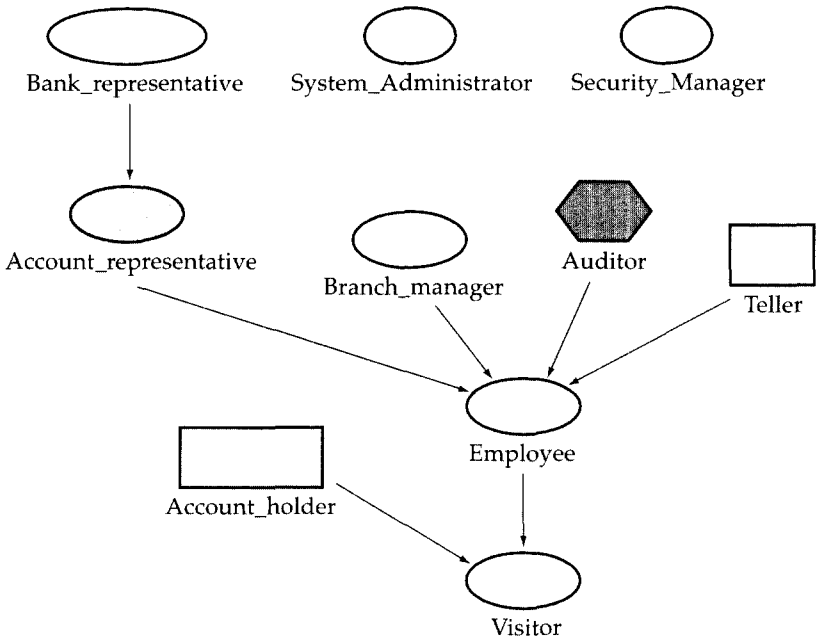
위와같이 각 직원들을 역할에 할당하면 그 직원은 할당된 역할에서 허가된 모든 작업들을 수행할 수 있다. 그러나, [표 1]에서 나타난 것처럼 국내 금융기관에서 발생한 금융사고 사례를 분석한 통계[8]를 보면 외부인에 의한 침해사고보다는 금융기관 내부직원에 의한 침해사고가 대부분이고 공격 유형자체도 단말기의 부정조작에 의한 것이다. 고객의 계정을 다루는 계정 책임자의 역할에 할당된 직원은 계정 내역을 조작할 수 있으며 살라미(SALAMI) 조작도 창구직원에 의해 수행가능하다. 이러한 조작을 방지하기 위해 3장에서 정의한 것과 같이 역할에 속성을 부가시킨다.

위에서 정의한 은행에서의 역할을 기반으로 [표 2]와 (그림 3)에서는 은행에 적용된 역할기반 접근통제에서의 허가와 속성을 표현한다.

[표 1] 금융기관 침해사고 유형

| 침해자 유형 | 분류기준 | 구성 비율 |
|---|---|---|
| 외부인 | 제3자에 의한 침해사고 | 17% |
| 내부인 | 내부직원에 의한 침해사고 | 81% |
| 외부인+내부인 | 내부직원과 공모에 의한 침해사고 | 2% |

(그림 3)과 같이 은행시스템에서 속성들의 집합은 {Time, Area, Amount, NoofTransactions}으




(그림 2) 은행에서의 역할 계층

[표 2] 역할에 부여된 객체에 대한 연산

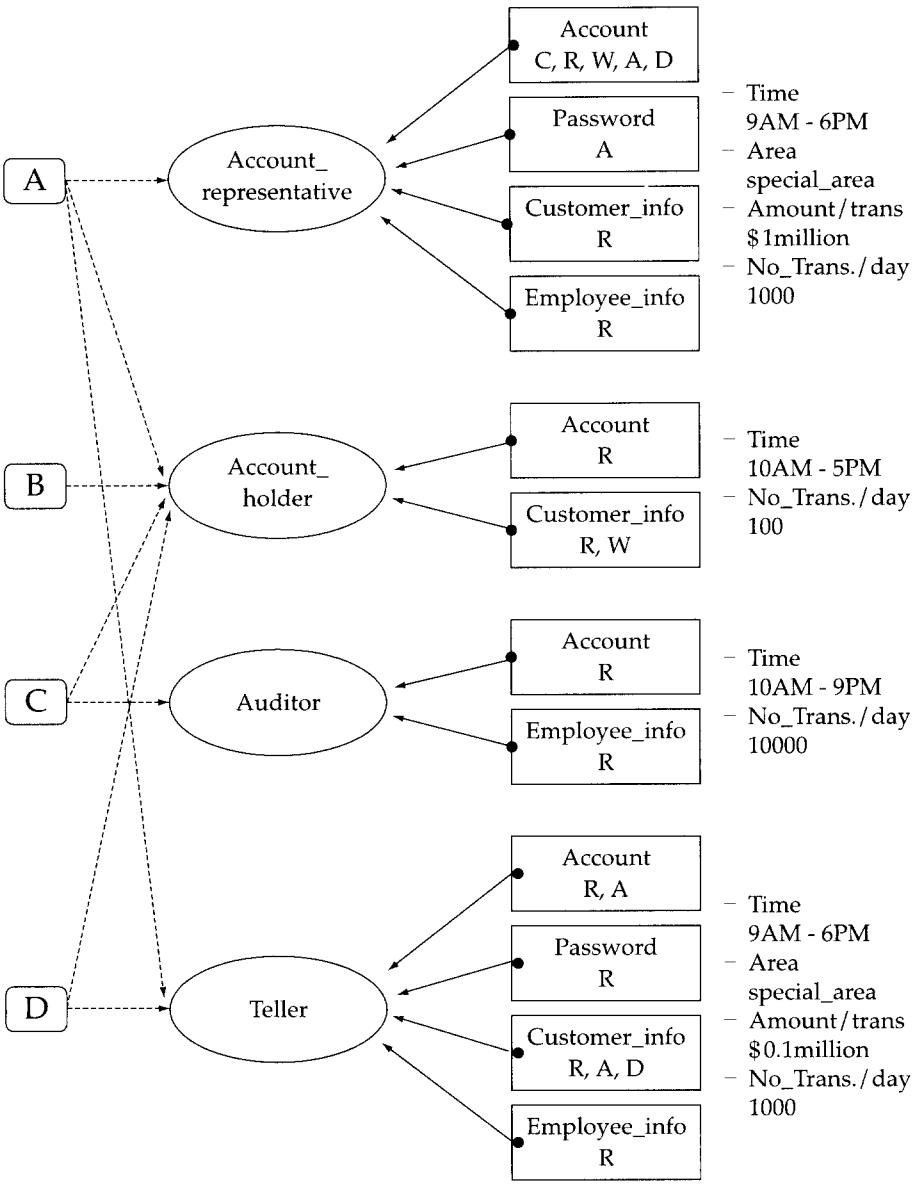
| 역할 | 계정 내역 | 패스워드 | 고객 정보 | 직원 정보 |
|---|---|---|---|---|
| 계정책임자 | C, R, W, A, D | A | R | R |
| 지점장 | R | A | R | C, R, W, A, D |
| 은행직원 | | | | R |
| 창구직원 | R, A | A | R, A, D | R |
| 계정소유자 | R | R, W | | |
| 내부 감사자 | R | | | R |

(C : 생성, R : 읽기, W : 쓰기, A : 추가, D : 삭제)

로 정의할 수 있으며 보안 관리자는 이 속성들 중에서 각각의 역할에 필요한 속성을 부가한다. 속성의 값은 다음과 같이 정의된다.

- TimeofWorks : ROLE -> TIME
  - TimeofWorks(r) : 역할 r을 수행할 수 있는 시간을 표시한다.
- AreaofWorks : ROLE -> AREA
  - AreaofWorks(r) : 역할 r을 수행할 수 있는 지역을 표시한다.
- Amount : ROLE -> AMOUNT
  - Amount(r) : 역할 r을 수행시 1회 처리 가능한 금액범위를 표시한다.
- No_of_Transactions : ROLE -> N
  - No_of_Transactions(r) : 역할 r에서 수행할 수 있는 처리횟수 제한을 표시한다.

수행하고 있는 역할이 가지고 있는 속성값은 다음과 같이 정의할 수 있다.




(그림 3) 은행에서의 역할기반 접근통제 모델

- ActiveTimeofWorks : ROLE -> TIME
  - ActiveTimeofWorks(r) : 역할 r이 수행되는 시간을 표시한다.
- ActiveAreaofWorks : ROLE -> AREA
  - ActiveAreaofWorks(r) : 역할 r이 수행되는 지역을 표시한다.
- ActiveAmount : ROLE -> AMOUNT
  - Amount(r) : 역할 r을 수행시 처리되는 금액을 표시한다.
- ActiveNoofTransactions : ROLE -> N
  - No_of_Transactions(r) : 역할 r로 수행되는 처리횟수를 표시한다.

현재 수행중인 역할에 부여된 속성 값이 정해진 속성 값보다 작거나 그 범위내에 있을
경우 객체에 대한 연산 권한을 가질 수 있다.

내부직원에 의한 부정조작 방지(The protection of attacks by insiders)

역할이 수행하는 속성 값이 정의된 속성 값을 벗어나지 않아야 이 역할에 할당된 허가가 수행될 수 있다. 즉, permitop(r,op,o,a)이 TRUE값을 돌려주어야 한다. 예를 들어 permit_op(Account_representative, Account, C, {PM2:00, Specialarea, $100, 25})은 TRUE 결과를 return한다. 속성 a는 PS(r, Attributes)의 한 원소로서 보안 관리자에 의해 각 역할의 특성에 따라 필요한 속성들이 정의될 수 있다. 은행의 경우 4개의 속성이 모두 정의되었을 때 다음과 같은 조건이 만족되어야 한다.

$\exists r \exists u, [r \in authorizedrole(u)] \wedge [ActiveTimeofWorks(r)] \in TimeofWorks(r)] \wedge [ActiveAreaofWorks(r) \in AreaofWorks(r)] \wedge [ActiveAmount(r) \leq Amount(r)] \wedge [ActiveNo\_ofTransactions(r) \leq NoofTransactions(r)]$

은행 내부 직원이 비정상행위(anomaly)를 시도하는 경우(예를 들어 정해진 근무시간 외의 조작, 한번에 많은 금액의 처리 등) 설정된 속성에 의해 이에 대한 허가 권한이 부여되지 않아 사전에 방지할 수 있다.

살라미 공격방지(Protection from Salami attack)

은행시스템에의 특정한 조작 방법인 Salami 공격을 방지하기 위해 다음과 같이 설정해 공격을 알아낼 수 있다. 이것은 아주 적은 금액(Salamiamount)을 많은 횟수(Salaminum) 조작하는 공격으로 특징적인 속성을 비교하여 방지한다.

$\forall r \exists u, [r \in authorizedrole(u)] \wedge [ActiveAmount(r) < Salamiamount] \wedge [ActiveNumofWorks(r) > Salaminum]$

물론 Salamiamount 및 Salaminum 값을 설정하기가 현실적으로 매우 어렵지만 통계자료등을 이용하여 각 은행의 실정에 맞게 적절한 값을 설정할 수 있다.

역할권한 집중방지(Decentralized authorization)

보안관리자(Securitymanager)는 권한이 한 역할에 집중되는 것을 원하지 않는다. [2]의 역할 계층 모델에서 정의한 것처럼 Bank representative는 Accountrepresentative의 역할 권한을 모두 상속받으나 이 경우 Accountrepresentative가 수행할 수 있는 모든 연산도 상속받게 된다. 이러한 연산의 상속은 상위 역할에 의한 부정 조작의 위험이 있으므로 상속받는 권한을 평상시에는 가지지 못하도록 해 권한을 상속하는 것에 대한 제한을 할 수 있다. 이것을 위해 differop 연산을 수행한다.

- differop(Bankrepresentative, Accountrepresentative) = authorizedperm(Bankrepresentative) - authorizedperm(Accountrepresentative)

이상과 같이 새로이 속성이 부가된 역할기반 접근통제 모델을 이용하여 은행시스템에 적용하였을 경우 기존의 모델이 가지고 있었던 내부직원에 의한 부정조작 위험 등을 방지할 수 있다. 은행이라는 특정한 목적을 위하여 시간, 지역, 금액, 횟수의 특정 속성을 부가시켜 이론적인 역할 기반 접근통제 모델에서 좀더 실세계에서 구현 가능한 모델을 제안하였다. 이 논문은 은행시스템에 대한 예를 보이고 있으나 정보 접근 통제를 구현하려는 다른 시스템에서도 효과적으로 사용될 수 있을 것이다.

## V. 결 론

역할기반 접근통제에 대한 최근의 많은 연구는 역할기반 접근통제가 상업적이고 행정적인 분야에서 유용하게 활용될 수 있을 것이라는 것을 보여준다 [9]. 개인 정보의 보호, 금융 자산에 대한 권한 없는 접근의 방지, 무단 정보 사용의 방지 등 다양한 분야에서 역할기반 접근통제가 적용될 수 있다.

그러나, 역할기반 접근통제를 실세계에 적용할 경우, 사용자가 권한내에서 부정 조작이 쉽다는 문제점과 역할계층상에서 상위에 존재하는 역할은 하위 역할의 연산을 상속받기 때문에 상위계층의 역할은 하위계층의 역할에서 가지고 있는 연산까지 모두 상속을 받게되어 부정조작 등의 위험성을 내포하고 있다.

본 논문에서는 이러한 문제점을 방지하기 위해 각 역할에 속성을 정의하였다. 정의된 속성은 시간, 지역, 트랜잭션의 횟수, 처리할 수 있는 돈의 액수 등이며 이러한 속성은 상호 결합하여 사용할 수 있도록 하였다. 이와 같이 정의된 역할 속성을 이용하여 이론적인 역할 기반 접근통제 모델을 실세계에서 구현 가능하도록 모델을 제안하였다. 또한 역할에 부여된 연산의 중복을 최대한 배제할 수 있도록 차이 연산자를 추가하였으며 이를 은행시스템에 적용한 예를 제시하였다. 본 논문에서는 제안한 역할기반 접근통제 방법은 접근통제를 구현하려는 다른 시스템에서도 효과적으로 사용될 수 있을 것이다.

초고속 정보통신망을 이용한 정보통신기반이 구축되고 있는 요즈음 역할기반 접근통제 방법을 금융, 물류, 에너지, 교통 등 다양한 정보통신기반 구조에 적용하기 위한 다양한 모델개발이 지속적으로 이루어져야 할 것이다.

## 참고문헌


[1] David F. Ferraiolo, Janet A. Cugini, and D. Richard Kuhn, "Role-Based Access Control(RBAC): Features and Motivations", 11th Annual Computer Security Applications Conference, Dec. 1995.

[2] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman, "Role-Based Access Control Models", IEEE Computer, Vol. 29, No. 2, Feb. 1996.

[3] Ravi S. Sandhu, Role-Based Access Control, Advances in Computers, Vol. 46, Academic Press, 1998.

[4] Luigi Giuri, "A New Model for Role-Based Access Control", 11th Annual Computer Security Applications Conference, Dec. 1995.

[5] Burkhard Hilchenbach, "Observations on the Real-World Implementation of Role-Based Access Control", 20th National Information Systems Security Conference, Oct. 1997.

[6] W.A.Jansen, "Ingeritance Properties of Role Hierarchies", 21th National Information Systems Security Conference, Oct. 1998.

[7] Barkley, Kuhn, Rosenthal, Skall, Cincotta, "Role-Based Access Control for the Web" ,CALS Expo International & 21st Century Commerce 1998: Global Business Solutions for the New Millennium

[8] 조이남, "금융정보망 정보보호현황 및 강화방향,, 제3회 정보보호 심포지움, 1998년 5월

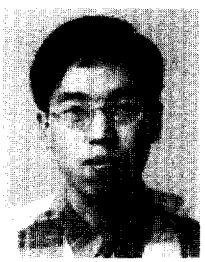
[9] David F. Ferraiolo, Dennis M. Gilbert, and Nickilyn Lynch, "An Examination of Federal and Commercial Access Control Policy Needs", 16th National Computer Security Conference, Sep. 1993.

## □ 著者紹介


**이 철 원**

1987년 2월 충남대학교 수학과(학사)
1989년 8월 중앙대학교 대학원 전산학과(석사)
1989년 9월 ~ 1996년 6월 한국전자통신연구소 선임연구원
1996년 6월 ~ 현재 한국정보보호센터 선임연구원

※ 주관심 분야 : 컴퓨터 · 네트워크 보안, 정보보호시스템 평가체제, 정보보호기술 표준화



**이 병 각**

1995년 2월 연세대학교(이학사, 전산과학과)
1997년 2월 연세대학교(공학석사, 컴퓨터공학과)
1997년 3월 ~ 현재 한국정보보호센터 기술개발부 연구원

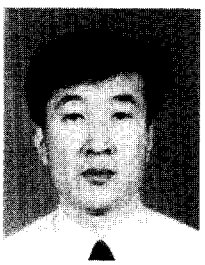
※ 주관심 분야 : 시스템 및 네트워크 정보보호

**김 기 현**

1993년 2월 경북대학교(공학사, 전자공학과)
1995년 2월 경북대학교(공학석사, 전자공학과)
1995년 7월 ~ 1996년 7월 데이콤 시외전화구축팀
1996년 7월 ~ 현재 한국정보보호센터 기술개발부 주임연구원
1997년 10월 ~ 현재 TTA 정보보호기술연구위원회 간사

※ 주관심 분야 : 시스템 및 네트워크 정보보호

## □ 著者紹介

박 정 호

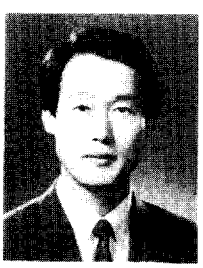
1984년 2월 한양대학교 산업공학과(공학사)
1986년 2월 한양대학교 산업공학과 대학원(공학석사)
1990년 7월 ~ 1992. 9. (주) 데이콤 연구원
1992년 10월 ~ 1996. 5. 한국전산원 선임연구원
1996년 5월 ~ 현재 한국정보보호센터 시험평가팀장

※ 주관심 분야 : 컴퓨터 · 네트워크 보안

이 홍 섭

1979년 2월 한양대학교(학사, 전자공학)
1985년 2월 한양대학교(석사, 전자공학)
1980년 ~ 1996년 한국전자통신연구소, 연구원 ~ 책임연구원 실장
1996년 ~ 현재 한국정보보호센터 기술개발 부장
정보통신기술협회 정보보호분과위원회 의장
한국통신정보보학회 상임이사

※ 주관심 분야 : 시스템 및 네트워크 정보보호



최 용 락

1976년 2월 중앙대학교 전자계산학과
1982년 2월 중앙대학교 전자계산학과 석사
1989년 2월 중앙대학교 전자계산학과 박사
1982년 ~ 1986년 한국전자통신연구원 선임연구원
1986년 ~ 현재 대전대학교 컴퓨터공학과 교수

※ 주관심 분야 : 운영체제, 분산처리체계, 컴퓨터통신보안