情報保護學會論文誌
第15券 第3號, 2005. 6

비밀성과 무결성을 보장하는 역할기반 접근제어모델*

변 창 우,† 박 석‡
서강대학교

A Role-Based Access Control Model ensuring Confidentiality and Integrity*

Chang-Woo Byun,† Seog Park‡
Sogang University

요 약

역할기반 접근제어의 중요한 특징은 그 자체가 정책 중립적이라는 데 있다. 이것은 역할기반 접근제어에는 특정한 접근제어 정책이 내포되어 있다기 보다 응용 환경에 따라 요구되는 정책을 쉽게 표현할 수 있다는 것을 의미한다. 이런 이유로 전통적인 접근제어 정책인 강제적 접근제어 정책과 임의적 접근제어 정책을 역할기반 접근제어 모델로 구성 가능함을 보이고자 하는 연구가 진행되어 왔다. 특히, 역할기반 접근제어를 이용하여 강제적 접근제어를 표현하는 연구에서는 낮은 보안등급에서 높은 보안등급으로의 단방향 정보흐름을 유지할 수 있는 두 가지 규칙인 하향 갱신 금지 규칙과 상향 판독 금지 규칙을 준수할 수 있도록 역할기반 접근제어의 일부 컴포넌트(사용자, 역할, 역할 계층, 세션)들을 재구성하고 제약사항들을 추가함으로써 해결하는데 초점을 두고 있다. 그러나 이런 기존 연구들은 비밀성 보장에 초점을 두었지만 실제 일부분에서 비밀성이 보장되지 못하고 있음을 밝힌다. 추가로 권한이 없는 사용자에 의한 정보 수정을 막는 무결성이 위반되고 있음을 밝힌다.

본 논문은 강제적 접근제어 정책에서 요구하고 있는 비밀성과 무결성을 동시에 만족시키는 역할기반 접근제어 모델을 제안한다. 역할기반 접근제어의 일부 컴포넌트들을 재구성하고 추가적인 제약사항들을 제안하였다.

ABSTRACT

An important characteristic of role-based access control model(RBAC) is that by itself it is policy neutral. This means RBAC articulates security policy without embodying particular security policy. Because of this reason, there are several researches to configure RBAC to enforce traditional mandatory access control(MAC) policy and discretionary access control(DAC) policy. Specifically, to simulate MAC using RBAC several researches configure a few RBAC components(user, role, role-hierarchy, user-role assignment and session) for keeping no-read-up rule and no-write-down rule ensuring one-direction information flow from low security level to high security level. We show these researches does not ensure confidentiality. In addition, we show the fact that these researches overlook violation of integrity due to some constraints of keeping confidentiality.

In this paper we propose a RBAC model satisfying both confidentiality and integrity. We reexamine a few RBAC components and constructs additional constraints.

Keywords : *role-based access control, mandatory access control, confidentiality, integrity*

접수일 : 2004년 12월 3일 ; 채택일 : 2005년 5월 6일
* 본 연구는 정보통신부 대학 IT연구센터 육성, 지원사업의 연구결과로 수행되었습니다.
† 주저자, chang@dblab.sogang.ac.kr
‡ 교신저자, spark@sogang.ac.kr

Ⅰ. 서 론

오늘날 조직에서 컴퓨터 환경을 갖춘 구성원들이 조직 내의 자원들을 접근하기 위해서는 적절한 접근 정책의 적용을 받아야 한다. 1985년 미 국방성에 의해 규정된 TCSEC(Trusted Computer System Evaluation Criteria)는 강제적 접근제어(Mandatory Access Control)와 임의적 접근제어(Discretionary Access Control)에 대해 규정하고 있다[1]. 강제적 접근제어는 군사 환경이나 매우 제한적인 환경에서 제한된 수의 보안 관리자들에 의해 일정한 규칙에 따라 사용자의 정보에 대한 접근을 통제한다. 반면에 임의적 접근제어 정책에서는 정보의 소유자들이 임의적으로 접근권한을 다른 사용자에게 위임할 수 있게 한다. 그러나 기업과 같은 상업적인 환경에서는 기업마다 서로 다른 보안 요구사항과 정책들을 반영해야 하기 때문에 강제적 접근제어나 임의적 접근제어만으로 이러한 요구를 만족시킬 수 없다[2].

R. Sandhu의 역할기반 접근제어 모델[3]은 기업과 같은 조직의 구조를 자연스럽게 반영할 수 있는 역할 구조를 지원하고 정책 중립적이라는 특징 때문에 기업마다의 서로 다른 보안 요구사항을 반영할 수 있고 권한 관리의 비용을 줄여주어 상업적인 응용에서 강제적 접근제어나 임의적 접근제어를 대체할 수 있는 접근제어 방법으로 좋은 평가를 받고 있다.

특히, 정책 중립적이라는 특징에 의해 자연적으로 강제적 접근제어 정책과 임의적 접근제어 정책을 역할기반 접근제어 모델로 표현할 수 있다는 기대와 함께 연구가 진행되어 왔다. 이런 연구가 이론적으로 중요한 이유는 강제적 접근제어 정책과 임의적 접근제어 정책을 역할기반 접근제어 모델로 표현함으로써 다른 동기로 다른 응용 환경을 위해 개발되었던 접근제어 모델들을 역할기반 접근제어 모델로 연관시킬 수 있기 때문이다. 한 예로 상업 환경의 응용에 다단계 보안 정책을 적용할 수 있게 된다는 것이다.

본 논문의 초점은 강제적 접근제어 정책의 비밀성 유지 규칙과 무결성 유지 규칙을 역할기반 접근제어 모델로 표현하는 것이다. R. Sandhu 와 S. Osborn에 의해 본 연구는 어느 정도 진행이 되어 왔다. 이들 연구는 역할기반 접근제어 모델이 무결성을 유지하는 정책이기 때문에 강제적 접근제어 정책의 상향 판독 금지(No-Read-Up)와 하향 갱신 금지(No-Write-Down) 규칙을 준수하는 역할 계층과 제약사항들에 초점을 두고 있다. 그러나 비밀성 규칙의 준수를 위해 제안한 제약사항들이 무결성을 위배한다는 사실을 간과하고 있다.

본 논문은 이런 무결성 위배 문제를 해결하면서 비밀성 규칙을 준수하는 새로운 역할 계층과 제약사항들을 갖고 있는 역할기반 접근제어 모델을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 강제적 접근제어 모델에서 비밀성을 보장하는 Bell-LaPadula 모델과 무결성을 보장하는 Biba 모델에서 요구하고 있는 규칙들을 살펴보고, 역할기반 접근제어 모델에 대해 간략히 설명한다. 3장에서는 강제적 접근제어 정책을 역할기반 접근제어 모델로 표현했던 기존 연구들을 살펴본다. 4장에서는 3장에서 소개된 기존 연구들이 갖고 있는 무결성 위배 문제 및 제약사항들의 문제점을 제시한다. 5장에서는 무결성의 문제점을 해결하고 비밀성을 보장하는 새로운 역할계층과 제약사항들을 갖고 있는 역할기반 접근제어 모델을 제시한다. 6장에서는 결론을 기술한다.

Ⅱ. 강제적 접근제어 모델과 역할기반 접근제어 모델

2.1 강제적 접근제어 모델

강제적 접근제어 모델에서 정보 보호에 대한 요구사항은 처리 대상인 정보의 속성에 따라 다양할 수 있지만 크게 두 가지로 구분된다. 그것은 비밀성(confidentiality)과 무결성(integrity)이다.

비밀성은 정보의 소유자 혹은 보안 관리자가 원하는 대로 정보의 비밀이 유지되어야 한다는 원칙이다. 정보는 비밀성이 노출되지 않도록 반드시 인가된 자에 의해서만 접근이 가능해야 한다. 무결성은 정보가 정해진 절차에 따라 주어진 권한에 의해서만 변경되어야 한다는 것이다. 정보는 항상 정확성을 유지하여야 하며, 인가 받은 방법에 의해서만 변경되어야 한다. 이러한 정보 보호의 기본적인 목표는 접근제어를 통해 이루어지게 된다.

어떻게 강제적 접근제어 모델에서 비밀성과 무결성을 보장하고 있는지 알기 위해 비밀성을 보장하는 Bell-LaPadula(이하 BLP이라 함) 모델과 무결성을 보장하는 Biba 모델을 간략히 소개한다.

· BLP 모델

BLP 모델[6]은 주체와 객체에 지정되는 비밀성을

위한 보안 등급(security level)에 근거하여 객체로의 접근을 제어함으로써 정보 보안의 목적 중 비밀성을 보장한다. 보안 등급의 구성 요소는 계층적 등급(hierarchical level)과 범주 집합(a set of categories)으로 이루어지고 계층적 등급은 전체 순서화(total ordering) 되어 있고 범주 집합은 부분 순서화(partial ordering) 되어 있다.

이러한 보안 등급들은 지배(dominate) 관계에 근거한 lattice를 구성하고 하나의 보안 등급 SL1이 다른 보안 등급 SL2를 지배한다는 것을 SL1 ≥ SL2로 표시한다.

이와 같은 BLP 모델은 비밀성 보장을 위해서 아래에 있는 보안 정책을 만족하는 경우만 객체로의 접근을 허용한다. 다음에서 SL(S), SL(O)는 각각 주체와 객체의 보안 등급이다.

BLP 모델에서 시행하는 보안 정책

1) Simple Security Property (No-Read-Up : NRU) : 주체 S의 보안등급이 객체 O의 보안등급을 지배하는 경우에만 판독 연산을 수행할 수 있다.
   조건 : SL(S) ≥ SL(O)
2) *-Property (No-Write-Down : NWD) : 주체 S의 보안등급이 객체 O의 보안등급에 지배되는 경우에만 갱신 연산을 수행할 수 있다.
   조건 : SL(S) ≤ SL(O)

2)의 경우 낮은 보안 등급의 주체가 높은 보안 등급의 객체에 갱신 연산을 수행할 수 있기 때문에 높은 보안 등급의 객체를 고의적으로 혹은 우발적으로 파괴하여 손실을 입힐 수 있게 되는 단점이 있어 갱신 연산에 보다 엄격한 규칙을 적용하기도 한다.

3) Strict *-Property (Write-Equal : WE) : 주체 S의 보안등급과 객체 O의 보안등급이 같은 경우에만 갱신 연산을 수행할 수 있다.
   조건 : SL(S) = SL(O)

BLP 모델은 정보의 보안만을 고려하기 때문에 정보가 낮은 보안 등급으로 흐르는 것을 막을 수 있지만 무결성을 보장하지는 못한다. 즉, 보안 등급이 낮은 사용자가 자신보다 상위 등급의 데이터에 쓰기를 수행할 수 있는데, 이 과정에서 상위 등급 데이터의 무결성이 깨어질 수 있다. 이러한 문제점을 개선한 모델이 Biba 모델이다.

· Biba 모델

Biba 모델[7]은 주체와 객체에 지정되는 무결성 등급(integrity level)에 근거하여 객체로의 접근을 제어함으로써 정보 보안의 목적 중에서 무결성을 보장한다.

하나의 무결성 등급 IL1이 다른 무결성 등급 IL2를 지배한다는 것을 IL1 ≥ IL2로 표시한다.

이와 같은 Biba 모델은 무결성 보장을 위해서 다음과 같은 보안 정책을 만족할 경우에만 객체로의 접근을 허용한다. 다음에서IL(S), IL(O)는 각각 주체와 객체의 무결성 등급이다.

Biba 모델에서 시행하는 보안 정책

1) No-Read-Down (NRD) : 주체 S의 무결성 등급이 객체 O의 무결성 등급에 지배되는 경우에만 판독 연산을 수행할 수 있다.
   조건 : IL(S) ≤ IL(O)
2) No-Write-Up (NWU) : 주체 S의 무결성 등급이 객체 O의 무결성 등급을 지배하는 경우에만 갱신 연산을 수행할 수 있다.
   조건 : IL(S) ≥ IL(O)

2.2 역할기반 접근제어 모델

역할기반 접근제어의 중심적인 개념은 사용자가 기업이나 조직의 정보 자원을 임의로 접근할 수 없도록 하는 것이다. 대신에 접근 권한이 역할(role)에 부여되고 사용자는 적절한 역할에 소속됨으로써 역할의 수행에 필요한 최소 자원만을 접근할 수 있도록 한다. 이러한 아이디어는 권한 관리를 매우 단순화 시켜주고 기업의 특정한 보안정책을 구현하는데 있어서 유연성을 제공하는 장점이 있다. 사용자는 그들의 업무적 권한과 책임에 따라 특정 역할의 구성원이 되며 접근 구조의 변경이 없이도 역할의 변경을 쉽게 할 수 있다. 그림 1은 RBAC의 기본 모델을 보여준다. RBAC의 기본 모델은 사용자(U: user), 역할(R: role), 인가권한(P: permission), 세션(S: session)으로 구성되어 있다[3-5].

사용자(user)와 역할(role) : 모델의 간략화를 위해서 사용자는 사람을, 역할(role)은 역할에 부여된 책임과 권한을 기술하는 조직내의 업무 기능(job function)의 이름으로 간주한다. 역할 계층(RH: role hierarchy)은 관련성이 있는 역할들 간의 부분순서(partial order) 관계로서 정의되며 기업의 권한과 책임의 체계와 매우 유사하여 기업의 권한체계를 모델링하는데 매우 적합하다.

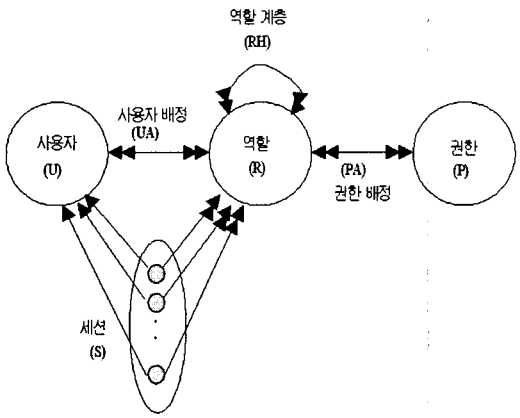


그림 1. RBAC의 기본 모델

인가권한(permission) : 인가권한은 시스템의 하나 또는 그 이상의 객체에 대한 특정 접근 모드(예 : read, write, update)의 승인을 나타낸다. 여기서 객체는 기업 또는 조직 내의 정보시스템을 구성하고 있는 자료나 시스템 자원을 말한다.

세션(session): 사용자는 시스템에 로그인(login)을 통해 그들이 가진 역할의 부분집합을 활성화할 때 세션을 형성한다. 각 세션은 하나의 사용자와 여러 개의 권한을 매핑(mapping)한다. 그림 1에서 이중 화살표는 다중 역할이 동시에 활성화한다는 것을 말한다. 사용자에게 사용 가능한 권한은 그러한 세션에 활성화된 모든 역할이 가진 권한의 합집합이다.

사용자 할당(user assignment)과 인가권한 할당(permission assignment) : 사용자 할당 및 인가권한 할당은 다대다 관계이며 RBAC 모델에서 매우 중요한 구성 요소이다. RBAC 모델의 특징 중의 하나는 사용자가 정보 객체들에 대해서 실행할 수 있는 연산들을 직접 사용자에게 부여하는 대신 조직의 업무 수행에 필요한 역할에 할당하고(인가권한 할당), 사용자는 해당 역할의 구성원이 됨으로써(사용자 할당) 정보 객체에 대해 지원하는 연산을 수행하도록 하는 것이다. 이러한 방법은 사용자와 정보 객체수가 많은 일반 기업 환경에서 권한의 관리를 매우 용이하게 수행할 수 있는 장점을 제공한다.

Ⅲ. 역할기반 접근제어 모델을 이용한 강제적 접근제어 모델 표현에 대한 기존 연구

강제적 접근제어 정책을 역할기반 접근제어 모델에 표현하는 연구는 두 그룹에서 집중적으로 연구되어 왔다.

강제적 접근제어 모델의 한 인스턴스인 lattice 기반 접근제어 모델을 역할기반 접근제어 모델로 표현하는 연구에서는[8,10] lattice 상의 정보 흐름은 상향 판독 금지, 하향 갱신 금지 규칙에 의해 높은 등급의 주체는 낮은 등급의 주체와 비교해서 판독 연산의 객체 범위는 크지만, 갱신 연산의 객체 범위는 작게 된다. 따라서 이런 lattice의 이중성을 역할 계층으로 소화하기 위해 판독은 위한 역할 계층과 갱신을 위한 역할 계층으로 나눠 처리하고 있다. 그림 2는 보안 등급 L 〈 (M1, M2) 〈 H를 가정했을 때의 판독 역할 계층과 갱신 역할 계층을 보여주고 있다.

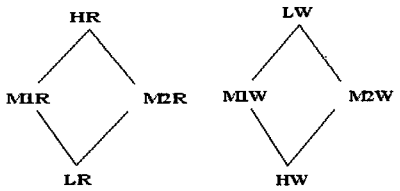


그림 2. 판독 역할 계층과 갱신 역할 계층 예

Simple Security Property와 *-Property :

사용자는 유일한 단일 보안 등급을 갖고 있다. 객체는 단일 보안 등급을 갖고 있다. 각 인가권한 쌍 (o, r)과 (o, w)은 정확히 하나의 xR과 xW에 할당된다. 이렇게 인가권한을 역할에 할당함으로써 간접적으로 객체는 자신의 보안등급을 갖게 된다. 단일 보안등급을 갖고 있는 사용자는 정확히 두 개의 역할인 xR 그리고 LW를 할당 받는다. 로긴 후 세션을 할당받을 때 사용자는 자신의 보안 등급에 지배되는 보안 등급에 해당되는 yR과 yW 역할을 할당 받는다($x \geq y$).

M1 보안등급을 갖고 있는 사용자는 판독 역할 계층에 의해 M1R 혹은 LR의 역할만을 할당 받을 수 있기 때문에 상향판독 금지를 준수하게 되며 갱신 역할 계층에 의해 M1W 혹은 HW의 역할만을 할당 받을 수 있기 때문에 하향 갱신 금지를 준수하게 된다.

Strict *-Property(Write Equal) :

Strict *-Property는 *-Property에서 갱신 역할 계층만 바꿔주면 된다. 그림 3 처럼 M1 보안등급을 갖고 있는 사용자는 판독 인가권한에 대해서는 *-Property와 같으며 갱신 역할 계층에 의해서는 M1W만을 할당 받을 수 있기 때문에 같은 보안등급

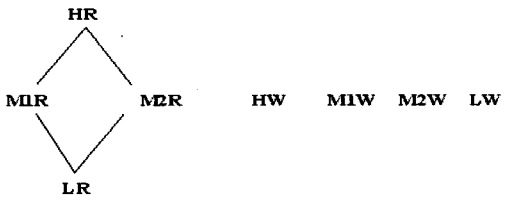


그림 3. Strict *-Property를 역할 계층 예

의 객체에 대해서 갱신 인가권한만을 갖게 된다.

위에서 언급한 역할 계층 구조는 다음의 상황을 만족하고 있지 못하다.

역할에 속한 권한 중 판독 데이터의 등급이 갱신 데이터의 등급보다 높은 경우 이들 권한들이 한 역할에 할당되면 이 역할보다 보안 등급이 높은 주체가 역할을 할당 받게 되어 하향 갱신(write-down)이 발생한다. 혹은 보안 등급이 낮은 주체가 역할을 할당 받으면 상향 판독(read-up)이 발생된다. 또한 예를 들어, M1의 등급을 갖고 있는 객체에 대한 판독 권한과 M2의 등급을 갖고 있는 객체에 판독 권한을 갖는 역할이 있다면 주체는 H 보안 등급을 가져야만 이 역할을 할당 받을 수 있는데 이를 해결하고 있지 못하다.

Osborn은 위에서 언급한 문제점을 두고 단일 역할 계층상에서 단일 보안등급을 갖고 있는 주체가 그 역할에 할당될 때 강제적 접근제어 정책의 두 가지 규칙인 상향 판독 금지와 하향 갱신 금지를 위반하는 지에 초점을 두고 변형된 역할 계층 및 제약사항들을 제시하고 있다[9,10].

일단 역할은 다른 등급의 인가권한들을 할당 받을 수 있음을 가정하고 있다. 이 가정에 의해 역할의 보안등급 범위 및 판독 인가권한의 최소 상계(least upper bound) 및 판독 인가권한의 최대 하계(greatest lower bound)를 가정하고 있다.

- r-scope(R) : 역할 R에 할당될 수 있는 모든 판독 인가권한들.
- w-scope(R) : 역할 R에 할당될 수 있는 모든 갱신 인가권한들.
- r-level(R) : 역할 R의 r-scope에 있는 객체 중 보안 등급의 최소 상계.
- w-level(R) : 역할 R의 w-scope에 있는 객체 중 보안 등급의 최대 하계.
- w-scope(R)이 공집합이면 암묵적으로 w-level (R)은 가장 높은 보안등급을 갖고 있음.
- r-scope(R)이 공집합이면 암묵적으로 r-level(R)은 가장 낮은 보안등급을 갖고 있음.

추가로 역할에 대한 제약사항 및 파생되는 역할 계층의 제약사항들을 제안하고 있다.

역할 생성제약사항 : 역할에 속한 w-scope의 최대 하계 w-level은 r-scope의 최소 상계 r-level을 지배한다(w-level(R) ≥ r-level(R)).

역할 계층 제약사항 1. 판독 전용 역할 계층 : 만약 판독전용 인가권한들만 갖고 있는 역할 R1과 R2에 대해서 역할 계층 R1 →[1] R2이 존재하려면 r-level(R2) ≥ r-level(R1) 이어야 한다.

역할 계층 제약사항 2. 갱신 인가권한이 포함된 역할 계층 : 만약 갱신 인가권한들을 포함한 역할 R1과 R2에 대해서 역할 계층 R1 → R2이 존재하려면

w-level(R2) =
greatest-lower-bound(w-level(R1),
w-level (w-scope(R2)−w-scope(R1))

이어야 한다.

위와 같은 역할 및 역할 계층의 제약사항들을 이용하여 강제적 접근제어 정책의 규칙들을 다음과 같이 설명하고 있다.

Simple security property :

상향 판독 금지 규칙을 보장하면서 임의의 주체 s가 역할 R을 할당 받기 위해서는 주체의 보안등급이 역할 R의 최소 상계를 지배해야 하다.

$\lambda^{2)}(s) \geq$ r-level (R)

***-property** :

하향 갱신 금지 규칙을 보장하면서 임의의 주체 s가 역할 R을 할당 받기 위해서는 주체의 보안등급이 역할 R의 최대 하계를 지배해야 한다.

$\lambda(s) \leq$ w-level (R)

Strict *-property :

상향 갱신 금지 규칙을 보장하면서 임의의 주체 s가 역할 R을 할당 받기 위해서는 역할 R의 모든 갱신 인가권한의 객체는 동일한 단일 보안등급을 갖고 있어야 하면 주체의 보안등급이 이 단일 보안등급과 같아야 한다.

Osborn은 추가적으로 세션에 대한 제약사항을 추가하였다[10].

1) R1 → R2의 의미는 역할 계층 상에서 R2는 R1의 상위 역할을 의미한다.

2) Osborn 모델에서 표기하고 있는 보안등급 기호

세션 제약사항 : 사용자의 보안등급은 형성된 세션의 보안등급을 지배해야 한다.

$((\forall s \in sessions, [\lambda(s) \leq \lambda(u)])$

특히, strict *-Propetry을 준수하기 위해서는 사용자의 보안등급은 형성된 세션의 보안등급과 같아야 한다.

$((\forall s \in sessions [\lambda(s) = \lambda(u)])$

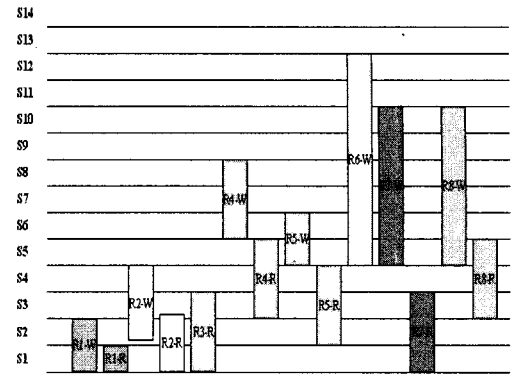
지금까지 살펴본 기존 연구들은 역할기반 접근제어 모델에 비밀성을 보장하기 위한 역할 계층 및 제약사항들에 대한 연구였다. 그러나 역할기반 접근제어의 순수 목적인 비권한 사용자에 의한 정보 노출 및 변경을 금지하는 무결성이 비밀성을 지키는 일부 제약사항들에 의해 위반된다는 사실을 간과하고 있다. 다음 장은 이 문제점에 대해 기술한다.

Ⅳ. 문제점

기존 연구의 문제점을 제시하기 위해 임의의 한 역할에 대한 판독 인가권한에 해당되는 객체들의 보안 등급 영역(r-scope(R)) 및 그 영역에서의 판독 인가권한의 최소 상계 (r-level(R))와 갱신 인가권한에 사용되는 객체들의 보안 등급 영역(w- scope (R)) 및 그 영역에서의 최대 하계(w-level(R))를 이용하여 임의의 역할 R1에서 R8을 구성한다. 그림 4.a는 역할 생성 제약사항에 의해 생성된 R1에서 R8까지의 각 역할에 대해 r-scope와 w-scope의 예제이고, 그림 4.b는 역할 계층 제약사항 1, 역할 계층 제약사항 2에 의해 생성된 역할 계층이다.

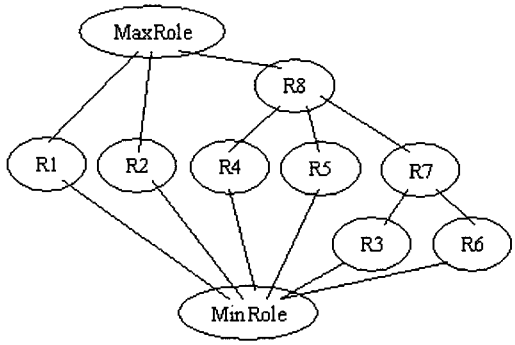
이와 같이 생성된 역할 계층, 세션 및 역할 계층에 존재하는 역할이 포함하고 있는 인가권한의 삽입·삭제에 의해 역할 계층을 재구성해야 하는 네 가지 문제들이 발생한다.

R1 : r-scope(S1), w-scope(S1~S2)
R2 : r-scope(S1~S2), w-scope(S2~S4)
R3 : r-scope(S1~S3)
R4 : r-scope(S3~S5), w-scope(S6~S8)
R5 : r-scope(S2~S4), w-scope(S5~S6)
R6 : w-scope(S5~S12)
R7 : r-scope(S1~S3), w-scope(S5~S10)
R8 : r-scope(S3~S5), w-scope(S5~S10)



a) R1에서 R8까지의 역할에 대한 r-scope 및 w-scope

MinRole :가장 낮은 보안등급을 갖고 있는 default 역할
MaxRole : 가장 높은 보안등급을 갖고 있는 default 역할
w-level(R3)≥w-level(R7) ∧
r-level(R7) ≥ r-level(R3)
⇒ R3 → R7
w-level(R7)=w-level(R5) ∧
r-level(R7)≥r-level(R6)
⇒ R6 → R7
w-level(R8)=w-level(R7) ∧
r-level(R8)≥r-level(R7)
⇒ R7 → R8
w-level(R8)=w-level(R5) ∧
r-level(R8)≥r-level(R5)
⇒ R5 → R8
w-level(R8)=w-level(R7) ∧
r-level(R8)≥r-level(R7)
⇒ R7 → R8



b)예제에 의해 생성된 역할 계층

그림 4. 문제점 제시를 위한 예제

〔문제점 1〕 역할 계층에 의한 정보 노출 문제

생성된 역할 계층에 의해 역할 R8에는 역할 R3, R4, R5 그리고 R7의 판독 권한들이 계승된다. 여기서 문제가 되는 것은 역할 R8의 r-scope에 해당되지 않는 판독 인가권한의 일부분(역할 R3, R5, R7의 일부 판독 인가권한)이 역할 R8에 계승되어 R8을 할당 받은 사용자가 권한이 없는 부분에 대한 판독을 행할 수 있게 되어 정보가 노출된다. 이것은 하향 판독이기 때문에 BLP 모델의 하향 판독 금지 규칙을 위반하지 않는다. 그러나 Biba 모델의 하향 판독 금지 규칙 즉, 판독 권한이 없는 사용자에게 판독이 허용되어 정보가 노출되는 문제점이 발생한다.

〔문제점 2〕 역할 계층에 의한 정보 수정 문제

생성된 역할 계층에 의해 역할 R8는 역할 R4, R5, R6 그리고 R7의 갱신 권한들이 계승되고 있다. 그러나 역할 R8의 w-scope에 해당되지 않는 역할 R6의 갱신 권한들이 계승되는 오류가 발생하고 있다. 이것은 실제 BLP 모델의 하향 갱신 금지 규칙을 위반하지 않는다. 그러나 Biba 모델의 상향 판독 금지 규칙 즉, 갱신 권한이 없는 사용자에게 갱신이 허용되어 정보가 수정되는 문제점이 발생한다.

〔문제점 3〕 세션에 의한 비밀성, 무결성 규칙 위반 문제

사용자 u의 보안 등급은 S5라고 하자. u가 할당받을 수 있는 역할들은 역할 R3, R4, R5, R6, R7 그리고 R8이다($[\lambda(u) \geq \text{r-level}(r)]$, $[\lambda(u) \leq \text{w-level}(r)]$).

세션의 제약사항($\forall s \in \text{sessions}\ [\lambda(s) \leq \lambda(u)]$)을 위반하지 않는 범위에서 이 사용자는 보안등급 S1, S2, S3, S4 혹은 S5로 시스템에 로그인하여 세션이 형성된다. 여기서 문제가 되는 것은 보안등급 S1 혹은 S2로 로그인 했을 때 허가된 역할 범위에서 벗어난 역할 R1 혹은 R2를 얻을 수 있어 BLP 모델의 하향 갱신 금지 규칙과 Biba 모델의 하향 판독 금지 규칙을 위반하게 되어 높은 보안등급에서 낮은 보안등급으로 정보가 흘러가는 정보흐름에 문제가 발생하고, 비권한 사용자에 의한 정보 노출 문제가 발생한다.

〔문제점 4〕 PRA 삽입·삭제에 의한 역할 계층 재구성

w-level(R)(또는 r-level(R))에 해당하는 갱신(판독) 인가권한의 삭제에 의해 역할의 보안등급 범위가 변경되고 이것은 곧 기존 역할 계층 구성에 변화를 초래한다. 예를 들어 역할 R8의 보안등급 S5의 갱신 인가권한을 삭제하여 w-level(R8)이 보안등급 S6이 되면 더 이상 역할 R8은 역할 R5, R6 그리고 R7의 상위 역할이 될 수 없다. 인가권한의 삽입에 의해서도 유사한 경우가 발생한다.

위에서 제시한 문제점들은 역할기반 접근제어 모델에서 기본적으로 제시하고 있는 역할 계층에서 하위 역할의 인간권한들이 상위 역할로 계승된다는 성질과 자신의 역할이나 하위 역할을 통해 세션을 형성할 수 있다는 성질에 비밀성을 위한 상향 판독 금지 및 하향 갱신 금지 규칙을 적용함으로써 발생된다. 또한, 역할 계층이 인가권한의 보안등급에 의존적이기 때문에 인가권한의 삽입·삭제에 의해 역할 계층에 대한 재구성이 발생된다. 다음 장에서는 이를 해결하기 위해서 필요한 역할 계층 및 세션에 대한 추가적인 제약사항들을 형식적인 표현을 제시한다.

V. 강제적 접근제어 모델을 만족하는 역할기반 접근제어 모델의 제안

일반적으로 역할기반 접근제어 모델은 역할을 할당받은 사용자가 그 역할에 할당된 인가 권한들을 이용할 수 있게 하여 비권한 사용자에 의한 정보 노출 및 수정을 금지하는 무결성을 준수하고 있다. 여기에 추가적으로 사용자에게 보안 등급이 주어지고 객체에 보안 등급이 주어져서 이들 간의 비밀성을 보장하도록 하기 위해서는 다른 보안등급을 갖고 있는 인가권한들로 구성된 역할에는 그 역할의 보안등급 범위 즉 판독 인가권한들의 최소 보안등급과 최대 보안등급, 갱신 인가권한들의 최소 보안등급과 최대 보안등급의 정보가 생성된다.

다음은 역할에 할당되는 인가권한들의 보안 등급을 구별하는 여섯 가지의 용어를 정의한다.

- r-scope(R) : 역할R에 할당될 수 있는 모든 객체의 판독 인가권한들의 집합.
- w-scope(R) : 역할 R에 할당될 수 있는 모든 객체의 갱신 인가권한들의 집합
- r-gub(R) : 역할 R의 r-scope에 있는 인가권한들 중 객체의 가장 높은 보안등급.
- r-glb(R) : 역할 R의 r-scope에 있는 인가권한들 중 객체의 가장 낮은 보안등급.
- w-gub(R) : 역할 R의 w-scope에 있는 인가권한들 중 객체의 가장 높은 보안등급.
- w-glb(R) : 역할 R의 w-scope에 있는 인가권한들 중 객체의 가장 낮은 보안등급.

5.1 사용자, 인가권한, 역할

사용자 정의 및 인가권한 정의의 형식적인 표현은 Osborn 모델[9,10]에서 제안하고 있는 형식적인 용어를 그대로 인용한다. 이에 대한 자세한 사항은 3장에서 언급하였다.

사용자 정의 : (∀U ∈ USERS), [SL[3](U) is given])

인가권한(P) 정의 : PERMISSIONS = {(o, r), (o, w) | o is an object in the system} ((∀(o,-) ∈ PERMISSIONS), [SL(o) is given])

역할(R) 제약사항 : 역할 R이 생성되기 위해서는 r-scope(R), w-scope(R), r-glb(R), r-gub (R), w-glb(R) 그리고 w-gub(R)이 설정되는데 생성된 역할은 반드시 그 역할의 갱신권한의 가장 낮은 보안등급은 판독권한의 가장 높은 보안등급을 지배해야 한다.

((∀R ∈ ROLES), [w-scope(R), r-scope (R), r-gub(R), r-glb(R), w-gub(R) and w-glb(R) are given])) w-glb(R) ≥ r-gub(R)

5.2 사용자-역할 할당(User-Role assignment: URA)

사용자-역할 할당(URA) 제약사항 : 생성된 역할들을 사용자가 할당 받기 위해서는 사용자의 보안등급이 역할의 판독 인가권한들 중 객체의 가장 높은 보안등급을 지배해야 한다. 한편, 역할의 갱신 인가권한들 중 객체의 가장 낮은 보안등급은 사용자의 보안등급을 지배해야 한다.

((∀(U, R) ∈ URA), [SL(U) ≥ r-gub (R)] ∧ [SL(U) ≤ w-glb (R)])

5.3 인가권한-역할 할당 (Permission-Role assignment : PRA)

처음 역할이 생성될 때 그 역할과 관련된 인가권한들이 할당되어 역할의 판독 인가권한의 보안등급 범위와 갱신 인가권한의 보안등급 범위가 정해진다. 그런 후, 생성된 역할에 새로운 인가권한의 삽입은 두 가지 대안을 생각할 수 있다. 첫째, 기존의 형성된 역할의 보안등급 범위를 만족하는 경우만 인가권한을 역할에 할당하는 방법이다. 본 논문에서 이를 〈PRA_1 삽입 제약사항〉이라 한다. 둘째, 삽입하고자 하는 인가권한의 보안등급을 허용하여 역할의 보안등급 범위를 새롭게 재구성하는 하는 방법이다. 본 논문에서 이를 〈PRA_2 삽입 제약사항〉라 한다.

추가로, 인가권한의 삭제 역시 두 가지 대안을 생각할 수 있다. 첫째, 기존의 형성된 역할의 보안등급 범위를 그대로 유지한 채 인가권한만 삭제하는 경우이다. 이 경우 어떠한 영향도 주지 않는다. 본 논문에서 이를 고려하지 않는다. 둘째, 삭제하는 인가권한에 의한 역할의 보안등급 범위를 새롭게 재구성하는 방법이다. 본 논문에서 이를 〈PRA_3 삭제 제약사항〉라 한다.

다음의 형식적인 표현들에 있어서 p_id는 삽입(혹은 삭제)하고자 하는 인가권한의 id이고, R은 p_id가 할당(혹은 제거)되는 역할이라 가정한다.

〈PRA_1 삽입 제약사항〉

단지 삽입하고자 하는 인가권한의 보안등급이 역할의 보안등급 범위에 속하는가만 확인하면 된다.

만약 판독 인가권한인 경우 :
SL(p_id) ≥ r-glb(R) ∧
SL(p_id) ≤ r-gub(R)
만약 갱신 인가권한인 경우 :
SL(p_id) ≥ w-glb(R) ∧
SL(p_id) ≤ w-gub(R)

이 방법을 사용한다면 새로운 인가권한의 삽입에 의한 역할의 보안등급 범위의 변경은 없기 때문에 역할 계층을 새롭게 구성할 필요가 없다.

〈PRA_2 삽입 제약사항〉

일단 삽입되는 인가권한의 보안등급을 역할 제약사항(w-glb(R) ≥ r-gub(R))을 만족해야 한다. 그림 5와 같이 세 가지의 경우를 생각해 볼 수 있다.

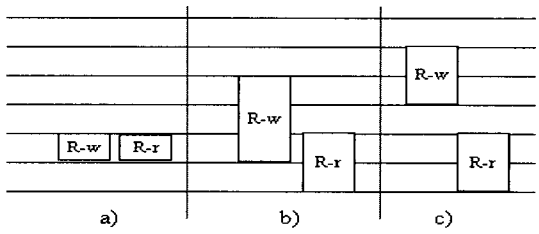


그림 5. 인가권한-역할 할당 제약사항 예제

3) 보안등급(Security Level)을 SL로 표현한다.

그림 5.a는 신뢰적인 주체(trusted subject) 및 비신뢰적인 주체(untrusted subject)가 객체에 대해 판독 연산과 갱신 연산을 요구하는 경우를 만족하는 경우이다.

그림 5.b는 비신뢰적인 주체의 정보 판독을 금지하는 비밀성을 유지하면서 신뢰적인 주체의 갱신 연산에 대한 무결성을 완화한 경우이다. 보안성을 손상시키지 않는다는 보장 아래 허용된 역할 보안등급 범위 하에 상향 갱신을 허용한다. 사용자의 보안등급이 갱신 인가권한의 가장 낮은 보안등급(이것은 판독 인가권한의 가장 높은 보안등급과 같다)과 같은 경우 신뢰적인 주체를 역할에 배정한다.

(SL(u)=w-glb(R)=r-gub(R))

그림 5.c인 경우는 BLP 모델에서 제안하고 있는 비신뢰적인 주체에 대한 판독 연산과 갱신 연산(여기서 말하고 있는 갱신 연산은 append 연산)을 허용하는 경우이다. append 연산이 발생되는 예는 로그 파일 작성과 같은 응용이 있다.

그림 5.a의 역할 보안등급 범위와 그림 5.b의 역할 보안등급 범위에서 삽입되는 인가권한은 다음의 경우에 허가된다. 이들 경우 다음과 같은 대등관계가 형성된다.

r-gub(R) = w-glb(R) = SL(U)
판독 인가권한인 경우 :
SL(p_id) ≤ r-gub(R)
갱신 인가권한인 경우 :
SL(p_id) ≥ w-glb(R)

이 경우 역할의 보안등급이 바뀌는 부분은 r- glb (R) 혹은 w-gub(R)이다. 따라서, 사용자의 보안등급과 역할의 r-gub(R)과 w-glb(R)의 지배관계에 의해 형성되는 사용자-역할 할당은 아무 영향을 받지 않는다. 즉, 역할의 보안등급 범위의 변경에 의한 사용자-역할 할당은 변하지 않는다. 또한, r-gub(R)과 w-glb(R)은 V.4절 역할 계층 형성 제약사항에서 비교 대상이기 때문에 역할계층 형성에 아무런 영향을 받지 않는다.

그림 5.c의 역할 보안등급 범위에서 삽입되는 인가권한의 허가 조건은 다르다. 일단 이 경우 다음과 같은 대소 관계가 형성된다.

r-gub(R) ≤ SL(U) ≤ w-glb(R)

따라서, 인가권한의 삽입에 의해 기존 사용자-역할 할당 관계의 파괴를 방지하기 위해 다음과 같은 제약사항이 필요하다.

판독 인가권한인 경우 :
((∀(U, R) ∈ URA), SL(p_id) ≤ SL(U)⇒
reconfigure_RH_by_addPRA(RRA, PRA, R)

갱신 인가권한인 경우 :
((∀(U, R) ∈ URA), SL(p_id) ≥ SL(U) ⇒
reconfigure_RH_by_addPRA(RRA, PRA, R)

이와 같은 제약사항을 만족하는 경우 역할의 보안등급 범위의 변경에 의한 사용자-역할 할당은 변하지 않는다. 그러나, V.4절 역할 계층 형성 제약사항에서 비교 대상인 r-gub(R), w-glb(R)의 변경을 초래하기 때문에 역할계층 재구성이 요구된다. 그림 6은 역할계층 재구성이 발생되는 두 가지 경우의 예를 보여주고 있다.

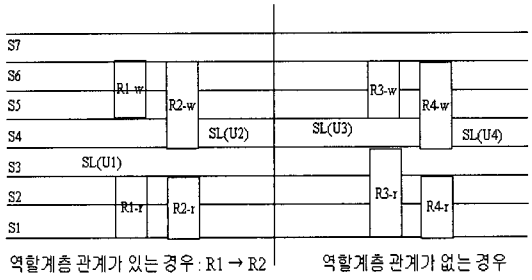


그림 6. 역할계층 재구성 예제

경우 1 : 역할계층 관계가 존재하는 경우
(U1, R1)∈URA, SL(U1)=S3,
r-glb(R1)=S1, r-gub(R1)=S2,
w-glb(R1)=S5, w-gub(R1)=S6
(U2, R2)∈URA, SL(U2)=S4,
r-glb(R2)=S1, r-gub(R2)=S2,
w-glb(R2)=S4, w-gub(R2)=S6

그림 6의 왼쪽 부분에서, r-gub(R1)=r-gub (R2)이고 w-glb(R1) ≥ w-glb(R2)이기 때문에 역할 R2는 역할 R1의 상위 역할이다. 예를 들어, 만약 역할 R1에 보안등급 S3인 갱신 인가권한이 삽입되면, 더 이상 역할 R2는 역할 R1의 상위역할이 될 수 없으며, 또한 다음 절에서 설명하고 있는 역할계층 생성 제약사항에 의해 역할 R1의 역할 R2의 상위역할이 된다.

경우 2 : 역할계층 관계가 없는 경우

그림 6의 오른쪽 부분에서, r-gub(R3) ≥ r-gub(R4)이지만 w-glb(R4) ≥ w-glb(R3)이기 때문에(반대 경우도 마찬가짐) 역할 R3와 R4는 역할 계층을 형성하지 않는다. 보안등급 S5를 갖고 있는 갱신 인가권한 이 역할 R3에 추가됨으로써 w-glb(R3)=S4로 변경된다. 이 경우 r-gub(R4) ≥ r-gub(R3)이면서 w-glb(R3)= w-glb(R4)이기 때문에 다음 절에서 설명하고 있는 역할계층 형성 제약사항에 의해 역할 R3은 역할 R4의 상위 역할이 된다.

그림 7은 이와 같이 인가권한의 삽입에 의한 역할계층 재구성을 해결하는 알고리즘을 보여주고 있다.

위에서 제안하고 있는 두 가지의 삽입 제약사항은 보안 정책의 선택사항이다. 〈PRA_1 삽입 제약사항〉은 한번 설정된 역할 보안등급을 절대 변경할 수 없도록 하는, 즉 인가권한-역할 할당이 한번 정해지면 역할의 보안등급 범위는 고정되고 그 고정된 범위 안에서 인가권한의 변경을 허용하는 환경에 적합하고, 〈PRA_2 삽입 제약사항〉은 역할의 보안등급이 사용자-역할 할당의 변경을 초래하지 않는 범위에서 어느 정도 인가권한의 삽입에 의한 역할의 보안등급의 변경을 허용하는 응용 영역에 적용되는 방법이다.

위에서 제안하고 있는 두 가지의 삽입 제약사항은 보안 정책의 선택사항이다. 〈PRA_1 삽입 제약사항〉은 한번 설정된 역할 보안등급을 절대 변경할 수 없도록 하는, 즉 인가권한-역할 할당이 한번 정해지면 역할의 보안등급 범위는 고정되고 그 고정된 범위 안에서 인가권한의 변경을 허용하는 환경에 적합하고, 〈PRA_2 삽입 제약사항〉은 역할의 보안등급이 사용자-역할 할당의 변경을 초래하지 않는 범위에서 어느 정도 인가권한의 삽입에 의한 역할의 보안등급의 변경을 허용하는 응용 영역에 적용되는 방법이다.

〈PRA_3 삭제 제약사항〉

인가권한의 삽입과는 반대인 경우로, 삭제될 인가권한의 보안등급이 갱신(혹은 판독) 인가권한의 객체의 가장 높은 보안등급과 가장 낮은 보안등급 사이에 존재한다면 삭제해도 아무 문제가 없다. 이와 같지 않은 경우 즉, 인가권한의 객체의 가장 높은 보안등급 혹은 가장 낮은 보안등급을 삭제하는 경우는 역할의 보안등급 범위가 변경되어 이에 파급되는 역할 계층의 재구성을 고려해야 한다. 삭제되는 인가권한에 의한 역할 계층의 재구성 알고리즘은 그림 8에서 보여주고 있다.

판독 인가권한의 삭제 경우 :
SL(p_id) = r-gub(R) or
SL(p_id) = r-glb(R) ⇒
Reconfigure_RH_by_rmPRA(RRA,PRA, R)

```
입력 :RRA, // 역할 계층 데이터 집합
 PRA, // 인가권한-역할 할당 데이터 집합
 r //삽입되는 인가권한을 할당 받는 역할
출력 :새로운 RRA 데이터 집합, 새로운 PRA 데 이터 집합
방법 :
Begin
/*삽입된 인가권한을 포함한 역할이 RRA에 속해 있지 않다면 역할 계층을 재구성할 필요가 없음. */
 if(r ∉ RRA)
 return abort
/*unchangeable RRA" 메시지를 보냄*/
 end if
 for all (p_r, r)∈ RRA do
/*p_r은 역할 r의 부모 역할*/
/* 새로운 인가권한을 할당 받은 역할이 상위 역할의 판독 인가권한의 객체의 가장 높은 보안등급보다 크거나 상위 역할의 갱신 인가권한의 객체의 가장 낮은 보안등급보다 작으면 역할 계층이 깨짐 */
 if((r-gub(p_r) ≤ r-gub(r)) ∨
 (w-glb(p_r) ≥ w-glb(r))
 then
 Reconfigure_RH_by_deleteRole(RRA,    PRA, r);
 Reconfigure_RH_by_addRole(RRA, PRA, r);
 end if
 /* 새로운 인가권한을 할당 받은 역할이 상위 역할의 판독 인가권한의 객체의 가장 낮은 보안등급보다 작거나 상위 역할의 갱신 인가권한의 객체의 가장 높은 보안등급보다 크면 계승 제약사항을 고려해야 함.*/
 if((r-glb(p_r) ≥ r-glb(r)) ∨
 (w-gub(p_r) ≤ w-gub(r)))
 then
 apply Rule of Limitation on read
 permission Inheritance or
 Rule of Limitation on
 write permission
 Inheritance
 end if
 end for
/*새로운 인가권한을 할당 받은 역할의 하위 역할을 찾을 필요는 없다. PRA 제약사항에 의해 역할 계층이 깨지는 경우는 발생하지 않음.*/
end.
```

그림 7. Reconfigure_RH_by_addPRA(RRA,PRA, r) 알고리즘

```
입력 : RRA, // 역할 계층 데이터 집합
 PRA, // 인가권한-역할 할당 데이터 집합
 r // 삭제된 인가권한에 대한 역할
출력 : 새로운 RRA 데이터 집합
방법 :
Begin
/*삭제된 인가권한에 대한 역할이 RRA에 속해 있지 않다면 역할 계층을 재구성할 필요가 없음.*/
 if(r ∉ RRA)
 return abort
 /*"unchangeable RRA" 메시지를 보냄*/
 end if
/*삭제된 인가권한과 관련된 역할의 하위 역할을 찾음.*/
 for all (r, c_r) ∈ RRA do
/* 삭제된 인가권한과 관련된 역할의 판독 인가권한의 객체의 가장 높은 보안등급이 하위 역할의 판독 인가권한의 객체의 가장 높은 보안등급보다 작거나 갱신 인가권한의 객체의 가장 낮은 보안등급이 하위 역할의 갱신 인가권한의 객체의 가장 낮은 보안등급보다 크면 역할 계층이 깨짐. */
 if((r-gub(r) ≤ r-gub(c_r)) ∨
 (w-glb(r) ≥ w-glb(c_r)))
 then
 Reconfigure_RH_by_deleteRole(
 RRA, PRA, r);
 Reconfigure_RH_by_addRole(
 RRA, PRA, r);
 end if
 /*삭제된 인가권한과 관련된 역할의 상위 역할을 찾을 필요는 없다. 역할 계층이 깨지는 경우는 발생하지 않음. 또한, 계승 제약사항도 고려할 필요는 없음.*/
 end for
end
```

그림 8. Reconfigure_RH_by_rmPRA(RRA, PRA, r) 알고리즘

갱신 인가권한의 삭제 경우 :
SL(p_id) = w-gub(R) or
SL(p_id) = w-glb(R) ⇒
reconfigure_RH_by_rmPRA(RRA,PRA, R)

5.4 역할 계층(Role Hierarchy : RRA)

역할 계층이 생성되기 위해서는 상위 역할의 판독 인가권한의 객체의 가장 높은 보안등급은 하위 역할의 판독 인가권한의 객체의 가장 높은 보안등급을 지배해야 한다. 또한, 상위 역할의 갱신 인가권한의 객체의 가장 낮은 보안등급은 하위 역할의 갱신 인가권한의 객체의 가장 낮은 보안등급에 지배되어야 한다.

역할 계층(RH) 형성 제약사항 :
$(\forall R_i, R_j \in ROLES)$ [w-scope(R_i), r-scope(R_i), r-gub(R_i), r-glb(R_i), w-gub(R_i), w-glb(R_i), w-scope(R_j), r-scope(R_j), r-gub(R_j), r-glb(R_j), w-gub(R_j) and w-glb(R_j) are given] $R_i \rightarrow^{4)} R_j \Rightarrow$ (r-gub(R_j) ≥ r-gub(R_i)) ∧ (w-glb(R_i) ≥ w-glb(R_j))

추가로 강제적 접근제어 모델의 비밀성과 무결성을 보장하는 역할의 의미를 부여하기 위해서는 하나의 역할에 할당된 다수의 다른 보안등급을 갖고 있는 객체에 대한 인가권한들에 대한 무결성을 만족시키기 위해 상위의 역할에 계승되는 인가권한들과 그렇지 않은 인가권한들을 구분해야 한다.

다음과 같은 $R_i \rightarrow R_j$ 역할 관계가 있다고 하자.

〈판독 인가권한 계승 제약사항: Limitation on read permission Inheritance〉

역할 R_j는 R_i의 모든 판독 권한을 계승하는 것이 아니라 r-glb(R_j)보다 보안 등급이 크거나 같고 r-gub(R_j)보다 작거나 같은 객체에 대한 판독권한들만이 계승된다.

$p_i \in$ r-scope(R_i) and
([SL(p_i) ≥ r-glb(R_j)] ∧
[SL(p_i) ≤ r-gub(R_j)])
$p_i' \in$ r-scope(R_i) and
[SL(p_i') 〈 r-glb(R_j)]

p_i 는 계승 가능한 권한이고, p_i'는 계승 불가능한 권한이다.

〈갱신 인가권한 계승 제약사항: Limitation on write permission Inheritance〉

역할 R_j는 R_i의 모든 갱신 권한들을 계승하는 것이 아니라 상속 받는 것이 아니라 w-glb(R_j)보다 보안 등급이 크거나 같고 w-gub(R_j)보다 작거나 같은 객체에 대한 갱신 권한들만이 계승된다.

$q_i \in$ w-scope(R_i) and
([SL(q_i) ≥ w-glb(R_j)] ∧
[SL(q_i) ≤ w-gub(R_j)])
$q_i' \in$ w-scope(R_i) and
[SL(q_i') 〉 w-glb(R_j)]

4) Ri → Rj 의 의미는 역할 계층 상에서 Rj는 Ri의 상위 역할을 뜻한다.

q_i는 계승 가능한 권한이고, q_i'는 계승 불가능한 권한이다.

한편 기존에 형성된 역할 계층 상태에서 역할의 삭제 혹은 새로운 역할의 삽입에 의한 역할 계층의 재구성을 고려해야 한다.

역할 삭제 : 삭제할 역할의 계승 가능한 인가권한들을 상위 역할에 계승한다. 그런 후, 역할을 삭제하고 새로운 역할 계층을 구성한다. 이에 대한 알고리즘은 그림 9에서 설명하고 있다.

역할 삽입 : 일단 삽입할 역할은 인가권한들을 할당 받고 있으며 그에 따라 보안등급 범위가 설정된다. 그런 역할을 기존 역할 계층에 삽입하기 위해서는 다음과 같은 두 가지 경우가 있다.

- 삽입할 역할이 상위 역할이 되는 경우 : 하위 역할의 인가권한들과 중복된 인가권한은 삽입한 상위 역할에서 삭제한다. 그런 후, 역할 계층을 재구성한다.
- 삽입할 역할이 하위 역할이 되는 경우 : 상위 역할의 인가권한들과 중복된 인가권한은 상위 역할에서 삭제한다. 그런 후, 역할 계층을 재구성한다.

이에 대한 알고리즘을 그림 10에서 설명하고 있다.

5.5 세션

사용자의 역할의 부분 집합을 활성화할 때 형성되는 세션에서도 비밀성을 유지해야 한다. 사용자의 보안등급은 형성된 세션의 보안등급을 지배해야 하고, 세션에 의해 활성화되는 역할은 사용자에게 할당 가능한 역할 중에 하나이어야 한다.

세션 제약사항 :

$(\forall U \in USERS), [SL(U) \text{ is given}])$

$((\forall s \in SESSIONS, [SL(s) \leq SL(U)] \wedge (\forall R' \in (\text{active-role}(s), (SL(s) \geq r\text{-gub}(R')] \wedge [SL(s) \leq w\text{-glb}(R')))$

active-role(s)은 사용자 U가 형성된 세션 s의 보안등급에 의해 할당 받는 역할 집합을 의미한다.

Ⅵ. 제안된 모델의 평가

접근제어 모델에 대한 평가는 그 특성상 성능평가나 정량적인 평가가 어렵다. 본 논문에서는 동일한 역할기반 접근제어 모델을 이용하면서 강제적 접근제어 정책을 표현한 기존 연구들과 비교 평가하여 어느 정도의 강제적 접근제어 정책에 필요로 하는 규칙들을 만족하고 있는가에 대해 기술한다.

먼저 기존 연구들과 비교해 보면 표 1과 같다.

```
사전 조건: ∀u ∈ USERS,
 n ∉ active-role(u)
입력 : RRA, // 역할 계층 데이터 집합
 PRA, // 인가권한-역할 할당 데이터 집합
 n //삭제할 역할
출력 : 새로운 RRA, PRA 데이터 집합
방법 :
Begin
/* 활동하고 있는 역할은 삭제할 수 없음.*/
 for all u ∈ USERS do
 if( n = active-role(u))
        then abort;
/*삭제하는 역할의 부모 및 자식 역할을 찾음.*/
 for all (r, n), (n, r') ∈ RRA do
/*r은 삭제할 역할의 부모 역할, r'은 자식 역할
*/
 if((n, p_id) ∈ PRA is inheritable)
 then
        insert (r, p_id) into PRA;
        delete (n, p_id) from PRA;
 end if
 delete(r, n) from RRA;
 delete(n, r') from RRA;
 insert(r, r') into RRA;
 end for
end.
```

그림 9. Reconfigure_RH_by_DeleteRole (RRA, PRA, n) 알고리즘

각 비교 항목별로 요약하면 다음과 같다.

- Sandhu 모델에서는 역할 생성에 있어서 역할에 속하는 인가권한들 사이의 보안등급 관계를 고려하고 있지 않다. 그러나, Osborn 모델에서는 역할 생성 제약사항(w-level(R) ≥ r-level (R)) 그리고 제안하는 모델에서는 역할 제약사항(w- glb(r) ≥ r-gub(r))을 두고 있기 때문에 역할 내에서 갱신 인가권한의 보안등급이 판독 인가권한의 보안등급을 지배하도록 하여 역할 내에서의 하향 갱신이나 상향 판독이 발생을 금지함으로써 비밀성을 유지한다.

```
입력 :   RRA, // 역할 계층 데이터 집합
         PRA, // 인가권한-역할 할당 데이터 집합
         n //삽입하는 역할, 보안등급이 설정됨.
출력 :   새로운 RRA 데이터 집합, PRA 데이터 집합
방법 :
ROLES role_set = ∅;
PERMS replication_privilege_set = ∅;
Begin
/*RRA 가 구성되어 있지 않다면 삽입되는 역할을 이용하여 RRA를 구성한다. */
        if(RRA = ∅)
          then
               insert (MAX_ROLE, n) into RRA;
               insert (n, MIN_ROLE) into RRA;
        end if
/*삽입하는 역할이 상위 역할이 되는 역할 계층의 하위 역할 집합을 찾음. */
        for all r ∈ RRA do
          if((r-gub(r) ≤ r-gub(n)) or (r-scope(r) = ∅))
            if((w-glb(r) ≥ w-glb(n)) or (w-scope(r) = ∅))
               then r ⊆role_set;
        end for
        for all r' ∈ role_set do
  if ((replication_privilege_set=(privilege_set(r') ∧ privilege_set(n))) ≠ ∅)
//RRA를 재구성하고, 인가권한 중복 해결
             for all (r'', r') ∈RRA do
                delete (r'', r') from RRA;
                insert RRA(n, r'); // n은 상위 역할, r'은 하위 역할
                insert RRA(r'', n); // r''은 상위 역할, n은 하위 역할
//중복된 인가권한 처리
        delete (n, p_id of replication_privilege_set) from PRA;
        if ((replication_privilege_set=(privilege_set(r'') ∧ privilege_set(n))) ≠∅)
          then delete (r'', p_id of replication_privilege_set) from PRA;
     end for
end for
/*삽입하는 역할이 하위 역할이 되는 역할 계층의 상위 역할 집합을 찾음.*/
        for all r ∈ RRA do
          if((r-gub(r) ≥ r-gub(n)) or (r-scope(r) = ∅))
            if((w-glb(r) ≤ w-glb(n)) or (w-scope(r) = ∅))
               then r ⊆role_set;
        end for
        for all r' ∈ role_set do
  if ((replication_privilege_set=(privilege_set(r') ∧ privilege_set(n))) ≠ ∅)
//RRA를 재구성하고, 인가권한 중복 해결
             for all (r', r'') ∈RRA do
                delete (r', r'') from RRA;
                insert RRA(n, r''); // n은 상위 역할, r''은 하위 역할
                insert RRA(r', n); // r'은 상위 역할, n은 하위 역할
//중복된 인가권한 처리
        delete (r', p_id of replication_privilege_set) from PRA;
        if ((replication_privilege_set=(privilege_set(r'') ∧ privilege_set(n))) ≠∅)
          then delete (n', p_id of replication_privilege_set) from PRA;
     end for
  end for
end
```

그림 10. Reconfigure_RH_by_InsertRole(RRA, PRA, n) 알고리즘

- Sandhu 모델에서는 명확히 역할의 보안등급 범위를 설정하여 사용자를 역할에 할당하는데 있어서 발생될 수 있는 비밀성을 위반할 수 있다. 그러나, Osborn 모델이나 제안하는 모델은 역할의 보안등급 범위를 설정하고, 역할에 할당하는 사용자의 보안등급이 역할의 판독 인가권한들 중 객체의 가장 높은 보안등급을 지배하고, 역할의 갱신 인가권한들 중 객체의 가장 낮은 보안등급에 지배돼야 하는 사용자-역할 할당 제약사항($(\forall (u, r) \in UA), [SL(u) \geq r\text{-}gub(r)] \wedge [SL(u) \leq w\text{-}glb(r)]$)을 둠으로써 역할에 사용자를 할당할 때 생길 수 있는 비밀성 위반을 예방한다.
- Osborn 모델을 기반으로 생성된 그림 4.b 역할 계층에서 인가권한들의 단순한 계승 원칙에 의해 역할 R7은 역할 R6의 상위 역할로서 R6의 모든 인가권한들을 계승할 수 있게 된다. 그러나, 역할 R6의 갱신 보안등급 범위는 보안등급 S5에서 S12까지이고 역할 R7의 갱신 보안등급 범위는 보안등급 S5에서 S10까지이다. 결국 역할 R7은 허용되지 않는 역할 R6의 보안등급 S11과 S12의 인가권한을 계승하는 무결성의 상향 갱신 금지 규칙을 위반한다. 제안하는 모델은 비밀성 기반으로 생성된 역할 계층에 갱신 인가권한의 계승 제약사항을 둠으로써 실제 계승되는 인가권한은 역할 R7의 보안등급 범위인 보안등급 S5에서 S10까지이며, 보안등급 S11과 S12에 해당되는 갱신 인가권한은 계승되지 않아 무결성의 상향 갱신 금지 규칙을 보장한다.
- 유사한 예로, 그림 4.b 역할 R8의 판독 보안등급 범위는 보안등급 S3에서 S5까지이며 역할 R7의 판독 보안등급 범위는 보안등급 S1에서 S3까지이다. 역할 계층에서 인가권한들의 단순한 계승 원칙에 의해 역할 R8은 역할 R7의 상위 역할로서 역할 R7의 모든 판독 인가권한들을 계승할 수 있게 된다. 결국 역할 R8은 허용되지 않는 역할 R7의 보안등급 S1과 S2의 판독 인가권한을 계승하는 무결성의 하향 판독 금지 규칙을 위반한다. 제안하는 모델은 비밀성 기반으로 생성된 역할 계층에 판독인가권한의 계승 제약사항을 둠으로써 실제 계승되는 인가권한은 역할 R8의 보안등급 범위인 역할 R7의 보안등급 S3 판독 인가권한뿐이며 역할 R7의 보안등급 S1과 S2의 판독 인가권한들은 계승되지 않아 무결성의 하향 판독 금지 규칙을 보장한다.
- 정적인 강제적 접근제어를 RBAC 모델로 시뮬레이션을 한 Sandhu 모델이나 Osborn 모델은 역할에 할당된 인가권한의 삽입·삭제에 대한 역할 계층의 재구성을 고려하고 있지 않다. 특히 Osborn 모델에서의 역할계층은 역할에 할당된 인가권한의 보안등급에 의해 역할의 보안등급 범위가 생성되고 그 역할의 보안등급 범위를 기반으로 생성되기 때문에 인가권한의 삽입·삭제에 의한 역할의 보안등급 범위가 변하는 것을 역할계층에 반영해야만 비밀성 및 무결성을 보장할 수 있다. 예를 들어, 그림 4.b에서 역할 R8의 기능을 변경하고자 보안등급 S5의 갱신 인가권한을 삭제하였다면 역할

표 1. 접근제어 모델 평가

	Sandhu 모델	Osborn 모델	제안하는 모델
역할 생성	Lattice 기반	인가권한 집합	인가권한 집합
역할 내에서의 비밀성 유지	X	O	O
사용자-역할 할당의 제약사항 지원	X	O	O
역할 계층에서 비밀성의 상향 판독 금지 규칙	O	O	O
역할 계층에서 비밀성의 하향 갱신 금지 규칙	O	O	O
역할 계층에서 무결성의 상향 갱신 금지 규칙	O	X	O
역할 계층에서 무결성의 하향 판독 금지 규칙	O	X	O
인가권한의 삽입·삭제에 대한 역할 계층의 재구성	N/A	N/A	O
역할의 삽입·삭제에 대한 역할 계층의 재구성	N/A	N/A	O
세션 내에서 무결성 지원	X	X	O
세션 내에서의 비밀성 지원	O	X	O

계층 생성 제약사항에 의해 더 이상 역할 R8은 역할 R5, R6 그리고 R7의 상위 역할이 될 수 없다. 역할계층을 그대로 유지한다면 역할 R5, R6 그리고 R7의 보안등급 S5에 해당되는 갱신 인가권한이 R8에 계승되어 무결성 규칙에 위배되고 판독 인가권한 측면은 비밀성 위반에 대해 유사하게 설명된다. 제안하는 모델은 〈PRA_3 삭제 제약사항〉을 두어 인가권한의 삭제에 의해 역할의 보안등급 범위가 변경됐을 때 역할계층을 재구성함으로써 이와 같은 문제를 해결하고 있다. 한편, 인가권한의 삽입에 의해서도 〈PRA-2 삽입 제약사항〉을 두어 유사하게 문제를 해결하고 있다.

- Sandhu 모델이나 Osborn 모델은 정적인 강제적 접근제어 모델을 RBAC 모델을 이용하여 시뮬레이션하였기 때문에 강제적 접근제어 모델의 문제점인 관리적인 측면을 배제하였다. 제안하는 모델은 강제적 접근제어 정책에 관리적인 요소를 추가하여 역할 계층을 구성하고 있는 역할의 삭제 및 새로운 역할의 삽입에 대한 역할 계층의 재구성을 통해 발생될 수 있는 비밀성 위반 상황 및 무결성 위반 상황 역시 통제할 수 있다.
- 그림 4.b의 역할계층을 기반으로 보안등급이 S5인 사용자 U가 할당 받을 수 있는 역할들은 역할 R3, R4, R5, R6, R7 그리고 R8이다($\lambda(u) \leq$ w-level(r)). 따라서, 세션의 제약사항($\forall s \in$ sessions $[\lambda(s) \leq \lambda(u)]$)을 위반하지 않는 범위에서 이 사용자는 보안등급 S1, S2, S3, S4 혹은 S5로 시스템에 로그인하여 세션이 형성할 수 있다. 여기서 문제가 되는 것은 보안 등급 S1 혹은 S2로 로그인했을 때 허가된 역할 범위에서 벗어난 역할 R1 혹은 R2의 역할을 얻게 되어 BLP 모델의 하향 갱신 금지 규칙과 Biba 모델의 하향 판독 금지 규칙을 위반하게 되어 높은 보안등급에서 낮은 보안등급으로 정보가 흘러가는 정보흐름에 문제가 발생하고, 비권한 사용자에 의한 정보 노출 문제가 발생한다. 제안하는 모델에서의 세션 제약사항($(\forall R' \in$ (active-role(s), $(SL(s) \geq$ r-gub(R')] $\wedge$ [SL(s) $\leq$ w-glb(R'))은 사용자의 보안등급이 세션의 보안등급을 지배하면서 활성화될 수 있는 역할은 사용자에게 할당 가능한 역할로 제한되기 때문에 역할 R1이나 R2는 사용자-역할 할당 제약사항에 의해 사용자 U가 절대로 할당될 수 없는 역할들이 된다.

Ⅶ. 결 론

일반적인 역할기반 접근제어 모델에 사용자의 보안등급과 객체의 보안등급을 부여하여 강제적 접근제어 정책을 표현하는 것은 다소 무리이다. 특히 역할기반 접근제어 모델에서 객체의 접근은 역할을 할당 받고 그 역할에 할당된 인가권한에 접근하고자 하는 객체가 속해 있어야만 한다. 여기에 비밀성을 보장하기 위해 상향 판독 금지 및 하향 갱신 금지 규칙을 적용하려면 접근에 필요한 요소인 역할 및 역할 계층, 인가권한의 비밀성 보장을 위한 제약사항들이 요구되며, 추가로 사용자-역할 할당 및 세션에 대한 제약사항들이 요구된다.

한편, 역할기반의 접근제어 모델에서 역할 계층이 갖고 있는 하위 역할의 인가권한들이 상위 역할로 계승된다는 특징 때문에 강제적 접근제어의 무결성을 보장하지 못하는 문제도 발생한다. 따라서 비밀성을 보장하면서 무결성을 보장할 수 있는 새로운 관점의 제약사항들이 필요하다. 추가로 상황에 따라 변경 가능한 역할기반 접근제어를 위해 인가권한의 삽입·삭제, 역할의 삽입·삭제에 의해 발생될 수 있는 비밀성 및 무결성에 대한 보장이 요구된다.

본 논문에서는 비밀성을 보장하면서 무결성을 보장하기 위해 관련 연구들에서 고려하지 않은 역할 계층에서 판독 인가권한 계승 제약사항과 갱신 인가권한 계승 제약사항을 통해 역할 계층에서의 인가권한들의 계승에 의해 발생되는 무결성 위반 문제를 해결하였고, 세션 제약사항을 통해 세션에 의해 발생될 수 있는 비밀성 및 무결성 위반 문제를 해결하였다.

추가로, 역할기반 접근제어 모델의 관리적인 측면에서, 인가권한-역할 할당 컴포넌트에서 인가권한의 삽입·삭제에 의한 역할의 보안등급 범위의 변경 및 이에 따른 역할 계층의 재구성에 대해서도 고려하고 있다. 또한, 역할 계층 컴포넌트에서도 역할의 삽입·삭제의 한 역할 계층의 재구성 및 그에 따른 인가권한의 재구성도 고려한 알고리즘을 제시하고 있다.

강제적 접근제어는 정적인 접근제어 특징을 갖고 있다. 그 이유는 기존 강제적 접근제어 모델에서는 관리적 측면을 고려하지 않고 있기 때문이다. 주체와 객체에 보안등급이 설정되고 설정된 보안등급에

따라 접근제어가 적용되는 환경은 주체와 객체의 민감성 계층을 구성하는 응용, 주체 혹은 객체의 분류에 의한 응용(부서, 프로젝트 작업) 등이 있을 수 있다. 이와 같은 응용 환경에서는 주체와 객체 사이의 관계를 변경할 수 있도록 허용하는 강제적 접근제어 모델이 필요하다. 따라서, RBAC의 권한관리의 효율성을 그대로 갖고 있으면서 BLP 모델과 Biba 모델을 지원할 수 있는 RBAC 모델을 연구하는 것은 관리적 측면을 고려한 강제적 접근제어 모델을 제안할 수 있는 방법이라 할 수 있다. 추가로 본 논문에서 말하고자하는 것은 강제적 접근제어 정책을 역할기반 접근제어 모델로 구현할 수 있다는 것보다 서로 다른 접근 정책을 하나로 표현할 수 있다는 것을 보여주고자 함이다.

한편, 추후 연구로 BLP 모델에서 제안하고 있는 비신뢰적 주체에 대한 갱신 연산인 append 연산을 순수하게 지원하기 위해 본 논문에서는 역할 내에서 발생하는 blind-write 문제를 허용하고 있다. RBAC 모델에서는 사용자를 역할에 할당하고 그 역할에 배정된 인가권한을 접근할 수 있게 하는 접근 메카니즘을 갖고 있는데, RBAC 모델에서 비신뢰적 주체의 의미를 어떻게 처리해야 할지는 명확하지 않다. 이 부분에 대한 연구를 통해 역할 내에서의 blind-write 문제를 해결해고자 한다. 추가로, 제안하는 모델에서의 많은 제약사항을 통해 실제 접근을 결정하는 응답시간과 처리율이 떨어진다. 제약사항을 통한 권한부여 관리에서 사용되고 있는 알고리즘의 성능을 분석하고 향상시킬 수 있는 방법에 대한 연구를 진행하고자 한다.

결론적으로 본 논문을 통해 단일 접근제어 모델(역할기반 접근제어 모델)로 비밀성과 무결성을 보장하는 강제적 접근제어 정책을 표현하여 서로 다른 접근 정책을 하나로 표현할 수 있다는 데에 의의가 있다고 볼 수 있다.

참 고 문 헌

[1] U.S. Department of Defense, Department of Defense Trusted Computer System Evaluation Criteria, *DOD 5200.28-STD, National Computer Security Center*, 1985.

[2] David D. Clark and David R. Wilson, "A Comparison of Commercial and Military Computer Security Policies", *Proc. of the 1987 IEEE Symposium on Security and Privacy*, 1987. pp.184-194.

[3] R. Sandhu, E. Coyne, H. Feinstein, and C. Younman, "Role-Based Access Control Models", *IEEE Computer Magazine Vol. 29*, 1996.2.

[4] C. Ramaswamy and R. Sandhu, "Role-Based Access Control Features in Commercial Database Management Systems", *NISSC*, 1998.

[5] R. Sandhu, "Role Activation Hierarchies", *Proc. of 3rd ACM Workshop on Role-Based Access Control*, 1998.10.

[6] D. E. Bell and L. J. LaPadula, "Secure Computer Systems: mathematical Foundations and Model", *Mitre Corp. Report No. M74-244*, Bedford, Mass., 1975.

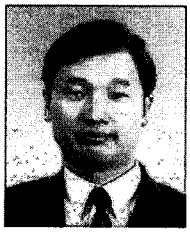
[7] K. J. Biba, "Integrity Considerations for Secure Computer Systems", *Mirte Corp. Report TR-3153, Bedford, Mass*, 1977.

[8] R. Sandhu, "Role-Hierarchies and Constraints for Lattice-Based Access Controls", *Proc. Fourth European Symposium on Research in Computer Security*, Rome, Italy, Sep. 25- 27, 1996.

[9] S. Osborn, "Mandatory Access Control and Role-Based Access Control Revisited", *Proc. of RBAC97, ACM*, 1997.

[10] S. Osborn, R. Sandhu and Q. Munawer, "Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies", *ACM Transactions on Information and Systems Security, vol.3, no. 2*, 2000.

〈著者紹介〉



박 석 (Seog Park)
1978년 2월 서울대학교 계산통계학과(이학사)
1980년 2월 한국과학기술원 전산학과(공학석사)
1983년 8월 한국과학기술원 전산학과(공학박사).
1983년 9월~현재 서강대학교 컴퓨터학과 교수
2002년~2004 University of Virginia 방문교수
1998년~현재 한국정보과학회 데이터베이스 연구회 운영자문위원
1997년 2월~현재 한국정보보호학회 이사
2004년 1월~현재 한국정보과학회 상임이사
2004년 1월~현재 한국정보과학회지 편집위원장
1999년~현재 DASFAA Steering Committee
2004년 DASFAA 2004 Organization Chair
〈관심분야〉 데이터베이스 보안, 트랜잭션 관리, 센서네트워크 데이터 관리, XML, 스트리밍 데이터 처리, 유비쿼터스 컴퓨팅, 역할기반 접근제어, 상황-인식 접근제어

변 창 우 (Chang-Woo Byun)
1999년 서강대학교 컴퓨터학과(학사)
2001년 서강대학교 컴퓨터학과(석사)
2001년~현재 서강대학교 컴퓨터학과 박사과정.
〈관심분야〉 Transaction Management for Dynamic Database, Role-based Access Control, Context-awareness Access Control