격자기반 역할그래프 보안 관리 모델

Role Graph Security Management Model based on Lattice

최 은 복* 박 주 기** 김 재 훈***
Eun-Bok, Choi Ju-Gi, Park Jae-Hoon, Kim

요 약

컴퓨터 시스템이 다양화된 분산시스템 환경으로 발전하면서 시스템에 존재하는 정보를 부적절한 사용자로부터 보호하기위한 접근통제 정책이 매우 중요하게 되었다. 본 논문에서는 강제적 접근통제모델의 등급과 역할기반 접근통제 모델의 제약조건과 역할계층을 체계적으로 변경함으로서 격자기반 역할그래프 보안 관리 모델을 제안한다. 이 모델에서는 기존의 역할그래프 모델의 역할계층에서 상위역할의 권한남용 문제를 해결하였으며 권한간의 충돌발생시 제약조건을 통해 주체의 등급을 재조정함으로서 정보의 무결성을 유지할 수 있다. 또한 역할계층에 의한 권한상속 뿐만 아니라 사용자의 보안레벨에 의해서 통제되도록 함으로서 강화된 보안기능을 제공한다. 그리고 본 모델을 운영하기 위해 역할그래프 보안 관리 알고리즘을 제시하였다.

Abstract

In this paper, we suggest lattice based role graph security management model which changes security level in mandatory access control model as well as constraint and role hierarchy systematically in role base access control model. In this model, we solved privilege abuse of senior role that is role graph model's problem, and when produce conflict between privileges, we can keep integrity of information by reseting grade of subject through constraint. Also, we offer strong security function by doing to be controlled by subject's security level as well as privilege inheritance by role hierarchy. Finally, we present the role graph algorithms with logic to disallow roles that contain conflicting privileges.

Key Words : MAC, Lattice, RBAC, Role Graph, Access Conrol

1. 서 론

1.1 연구 배경

컴퓨터 시스템이 여러 사용자나 여러 주체가 여러 개의 응용프로그램을 동시에 사용하는 다양화된 분산시스템 환경으로 발전하면서 시스템에 존재하는 정보를 부적절한 사용자로부터 보호하기위한 정보 보안이 매우 중요하게 되었다. 이에 따라 시스템 관리자와 보안 프로그램의 개발자들은 허가 받은 사용자나 주체만이 자원에 접근할 수 있도록 통제하는 접근통제 기법에 관심을 가지게 되었다.

접근통제 정책에는 다음과 같이 크게 세 가지 정책으로 나눈다.

자율적 접근통제(Discretionary Access Control)정책은 접근을 요청한 주체가 객체에 대한 권한을 자율적으로 다른 주체에게 권한을 부여하거나 철회할 수 있음을 의미하며, 종류로는 접근통제행렬(Access Control Matrix), 접근통제리스트(Access Control List), 능력리스트(Capability List) 등이 있다. 강제적 접근통제(Mandatory Access Control)정책은 시스템 관리자에 의해 보안등급이 결정되는 정책으로 대표적인 모델로는 정보의 비밀성을 중

* 정 회 원 : 전주대학교 정보기술공학부 조교수
ebchoi@jj.ac.kr
** 정 회 원 : KT 책임연구원
jugipark@kt.co.kr
*** 정 회 원 : 전주대학교 교양학부 객원교수
muggeby@jj.ac.kr
[2006/07/03 투고 - 2006/08/01 심사 - 2006/09/04 심사완료]

요시하며 군사 분야의 응용에 적합하도록 만들어진 *BLP* 모델과 정보의 비밀성보다는 무결성을 강조하기 위한 Biba 모델이 있다[1].

역할기반 접근통제(Role-Based Access Control) 정책은 역할에 기반을 두고 시스템 자원에 대한 사용자의 접근을 제어하는 것으로, 각 역할에 미리 정해놓은 접근권한을 부여하고 사용자에게 역할을 부여함으로서 사용자가 가질 수 있는 접근권한을 통제한다. 역할 기반 접근통제에서 역할은 기업이나 응용 프로그램 환경에서 의미있는 권한들의 집합으로 사용되며, 많은 사용자와 많은 자원들을 갖는 복잡한 시스템에서 업무 수행에 필요한 권한을 관리하는데 적절한 기능을 제공하므로 기업 관리자나 보안 관리자에게 다양한 작업 기능의 권한을 기술하는데 강력한 방법을 제공한다[2,3].

1.2 연구 필요성

분산시스템 환경에서 접근통제에 의해 유지되는 관리객체는 시스템 환경 중 특히 네트워크 관리 시스템의 핵심적인 부분으로서 망 관리에 필수적이며 중요한 모든 정보를 유지하고 있기 때문에 안전하게 유지되어야 한다. 이러한 네트워크 관리 정보를 안전하게 운용되기 위해서는 네트워크 사용자에 대한 정확한 인증 뿐만 아니라 관리객체에 대한 접근을 효율적으로 통제할 수 있어야 한다. 특히, 상호독립적으로 운영되는 통신망들이 상호연동 됨에 따라 전체적인 통신망의 규모가 점점 커지고 복잡해지고 있으며 다양한 사용자들로부터 정보를 안전하게 유지관리되기 위해서는 의도적인 정보 변경을 방지하는 무결성이 보장되어야 한다.

상업적인 측면의 보안 정책을 강화하는 역할기반 접근통제 정책은 사용자들이 수행하는 공통적인 기능들에 기반을 둔 그룹들인 역할로 구성되며 조직이나 환경에 따라 역할이 자연스럽게 생성되고 재구성될 수 있는 유연성을 갖는다. 그렇지만, 역할계층의 경우 부분순서 지배관계에 따라 상위역할의 경우에 하위역할의 모든 권한을 상속받으므로 권한간의 충돌 및 남용 문제가 발생된다. 또한, 역할을 수행하는 관리자와 관리객체에 대한 접근통제 방법이 제공되지 않아 정보의 무결성을 해칠 우려가 있다.

역할기반 접근통제 정책에서는 이러한 문제점을 보완하기위해 임무분리(separation of duty) 특성을 구현하기 위한 방법들 중 하나로 상호 배타성(mutual exclusiveness)이 있다. 이는 정보의 무결성에 손상을 입힐 수 있는 권한을 가진 역할들, 즉 상호 배타적 역할들이 수행할 수 있는 권한들 간의 충돌 여부에 따라 이 역할들이 동일 사용자에게 할당하지 않음으로써 무결성 침해의 가능성을 최소화한다. 그러나 상호 배타적 역할의 지정에 대한 일반적인 규칙은 존재하지 않고 응용 프로그램의 문맥에 따라 다르므로 각 시스템의 보안 관리자에 의해 임의로 지정되고 있다.

강제적 접근통제 모델의 문제점은 권한을 갖지 않는 사용자에게 정보가 흐르는 것을 방지하는 정보의 비밀성을 보장하는 데는 효과가 있지만 인가등급이 낮은 주체가 보안등급이 높은 객체를 부당하게 훼손하거나 변경할 우려가 있어 정보의 무결성을 보장하지는 못한다는 것이다.

그러므로 본 논문에서는 정보의 이용가능성 측면에서 정보의 비밀성보다는 정보의 무결성을 보장하기위한 보안정책을 적용하였으며 상업적인 실생활에 적용 가능하도록 역할기반 접근통제 정책을 사용하였다. 특히, 역할계층의 부분순서 지배관계에 따라 상위역할의 경우에 하위역할의 모든 권한을 상속받으므로 상위 역할의 권한 남용 문제를 해결하기 위해, 관리업무 수행의 책임권한과 관리정보의 중요도에 따른 주체와 객체의 인가 및 보안등급을 적용하는 격자기반 역할 그래프 보안 관리 모델을 제안하였다.

정보 통신망의 관리 정보를 이용하는 사용자의 환경이 동적으로 변화하는 현대의 네트워크 환경에서 다양한 접근통제 정책들 간의 연관성의 연

구가 필요하다고 볼 때, 본 논문은 실용적인 정보의 이용성과 무결성을 동시에 보장하기 위한 접근통제 정책들의 상호연관성 연구에 기여하리라 본다.

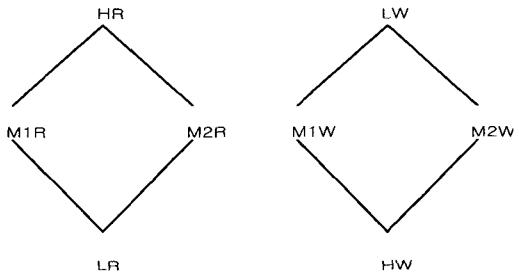
2. 관련연구

2.1 격자기반 접근통제 모델

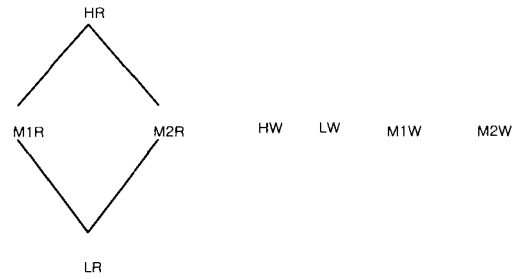
격자기반 접근통제는 보안등급에 기반하여 일방향의 정보흐름을 강조하는데 관점을 둔 강제적 접근통제 정책의 한 모델로서, 사용자나 주체에 부여된 인가등급(security clearance)과 객체에 부여된 보안등급(security classification)을 적용하여 접근통제 여부를 결정한다[4].

격자기반 접근통제 모델은 (그림 1)과 (그림 2)와 같이 비밀성에 기반을 두므로 정보의 흐름이 아래에서 위로 향하는 구조를 갖으며 부분적인 순서지배관계를 갖는 보안등급의 유한격자구조를 갖는다. 이 그림에서 H등급과 L등급은 보안등급의 높고 낮음을 뜻하며 M1과 M2등급은 H등급과 L등급의 중간으로 등급의 비교가 불가능한 경우를 의미한다.

격자기반 접근통제 모델은 다음과 같이 크게 두 가지 특성을 갖는다. 정보의 읽기에 관련된 단순한 보안 특성과 쓰기에 관련된 스타 특성으로 나뉜다.



(그림 1) 격자구조-단순한 및 관대한 보안특성



(그림 2) 격자구조 - 단순한 및 엄격한 보안특성

▷ 단순한 보안 특성(Simple Security Property)

주체 s의 인가등급이 객체 o의 보안등급에 부분순서지배관계이면 객체 o를 읽을 수 있다.

IF $\lambda(s) \geq \lambda(o)$ THEN read(s,o) ELSE reject;

▷ 관대한 스타 특성(Liberal *-Property)

객체 o의 보안등급이 주체 s의 인가등급에 부분순서지배관계이면 주체 s는 객체 o를 쓸 수 있다. IF$\lambda(s) \leq \lambda(o)$ THEN write(s,o) ELSE reject;

관대한 스타 특성은 낮은 주체가 높은 등급의 객체를 쓸 수 있기 때문에, 무결성 측면에서 보면 높은 등급의 데이터가 의도적이던지 사고에 의하던지 낮은 등급의 주체에 의해 파괴되고 훼손될 수 있음을 의미한다. 이러한 가능성을 피하기 위해 다음과 같은 엄격한 스타 특성이 적용된다.

▷ 엄격한 스타 특성(Strict *-Property)

주체s의 인가등급이 객체 o의 보안등급과 같을 경우에 객체 o를 쓸 수 있다.

IF $\lambda(s) = \lambda(o)$ THEN write(s,o) ELSE reject;

2.2 역할기반 접근통제 모델

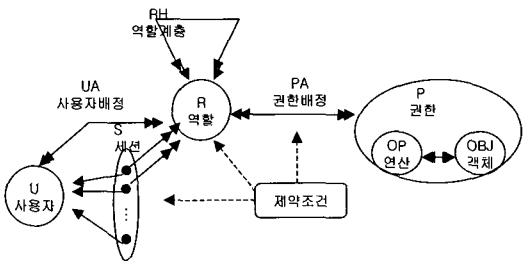
역할기반 접근통제 모델은 (그림 3)과 같이 사용자(U), 역할(R), 권한(P)이라 불리는 세 개의 개체의 집합에 기반을 둔다. 역할은 권한들의 집합으로써 역할의 이름인 name과 역할이 수행하는

권한들의 집합인 rpset의 집합인 (rname, rpset)으로 구성된다. 역할간의 관계는 역할계층에서 표현되며 r_i.rpset⊂r_j.rpset 관계가 형성되면 역할 r_i의 모든 권한을 역할 r_j가 수행할 수 있음을 의미하며 r_i is-junior r_j 또는 r_j is-senior r_i라고 표기한다[5,6].

권한은 객체 x와 객체에 대한 접근모드 m을 갖는 (x, m)의 순서쌍으로 구성된다. 여기서 m은 간단한 시스템의 경우 read, write, execute와 같은 접근모드이거나 객체지향 환경의 경우 메소드중 하나의 실행모드일 수 있으며 객체 x의 접근을 용이하게 하는 하나의 트랜잭션일 수 있다. 객체 x나 접근모드 m의 정확한 특성은 응용환경이나 그들에 연관된 보안정책에 따라 다양하게 쓰일 수 있다.

사용자배정과 권한배정은 다대다 관계를 갖는데, 한명의 사용자는 많은 역할들에 배정될 수 있으며 하나의 역할이 많은 사용자에 배정될 수 있음을 의미한다. 또한, 하나의 역할에 많은 권한이 배정되고 동일한 권한이 많은 역할에 배정될 수 있다.

부분순서쌍을 갖는 역할계층은 '≤'으로 표기하며 여기에서 x ≤ y의 의미는 역할 y가 역할 x에 배정된 권한을 상속받는 것을 뜻한다. 역할계층은 비순환방향성 그래프로서 방향성 관계는 에지로 표현되며 상속관계는 이행성을 갖는다. 역할계층은 역할과 역할의 부분순서 관계인 이진관계 (RH ⊆ R X R)로 표현된다. 사용자와 권한은 각각 이진관계 UA ⊆ U X R과 PA ⊆ P X R에 의하여 역할과 연관되며 U는 사용자의 집합이며 P는 권한의 집합이다.



(그림 3) 역할기반 접근통제 모델

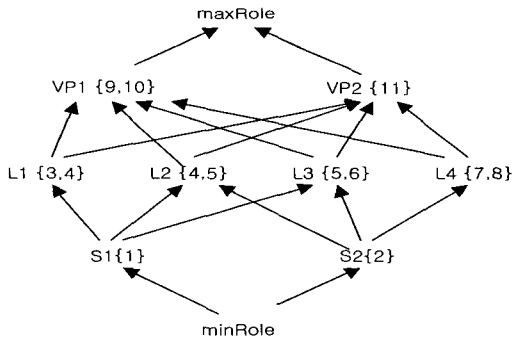
각 세션은 하나의 사용자와 연관이 되며 사용자는 하나의 시스템에서 서로 다른 윈도우를 열어 동시에 다중 세션을 열 수 있다. 세션의 개념은 전통적인 접근통제에서 주체의 개념과 동일하다. 주체나 세션은 접근통제의 단위이며 하나의 사용자는 동시에 다른 권한으로 구성된 여러 개의 주체나 세션을 가질 수 있다.

제약조건은 다양한 컴포넌트에도 적용될 수 있는데, 구매 관리자와 지불관리자 역할처럼 상호배타적인 역할의 경우 동일한 사용자에게 두 가지 역할이 동시에 배정되지 않도록 제약조건이 기술되어야 한다[7,8,9].

2.3 역할그래프

역할그래프는 비순환 방향성을 갖는 그래프로서 시스템에서 역할을 표현하는 노드와 노드간의 부분순서관계인 'is-junior' 관계를 표현하는 에지로 구성된다. 역할 그래프는 최대역할과 최소역할을 가지며 최대역할은 역할그래프에서 역할들의 모든 권한의 합집합이며 최소역할은 모든 역할들이 이용가능한 최소한의 권한집합을 갖는다. 역할그래프에서 r_i is-junior r_j이면 r_i에서 r_j로 가는 경로가 존재함을 의미하고 r_i의 모든 권한은 r_j를 통해 이용가능하며 이는 r_j에 배정된 사용자는 역할 r_i의 모든 권한이 이용가능함을 의미한다[10].

역할그래프는 상하위 역할간의 상호작용을 시각



(그림 4) 역할그래프의 예

화한 모델로 사용자, 역할, 권한의 세 가지 요소로 구성된다. 사용자는 시스템의 자원을 사용하는 개체로서 역할 그래프에서는 그룹이라는 사용자들의 집합을 만들어 사용자들에 대한 일괄적인 관리도 가능하게 하고 있다. 권한은 시스템의 정보나 자원에 대한 특정 작업의 승인으로서, 접근하려는 객체와 그에 대한 작업의 쌍으로 이루어진다. 역할그래프에서는 권한을 다시 실제권한(effective privilege)과 직접권한(direct privilege)으로 구분하였다. 역할 r의 직접권한은 역할 r에 대한 하위 역할의 rpset에 포함되지 않는 권한으로 자신만의 고유 권한을 의미하며, 실제권한은 역할 r의 하위역할들의 실제권한과 직접권한의 합집합에 해당된다. (그림 4)에서 역할의 오른쪽 괄호로 표현된 것이 그 역할의 직접권한이며 그림에 대한 직접권한과 실제권한은 (표 1)에서 정리하였다.

3. 격자기반 역할그래프 보안 관리 모델

격자기반 접근통제모델은 주체의 인가등급과 객체의 보안등급에 의해 정보의 합법적인 접근여부가 결정되는 정책으로 일방향성을 갖는 정보의 흐름을 통해 정보의 비밀성을 중요시하는 모델이다. 이러한 정책 모델은 권한을 갖지 않는 사용자에게 정보가 흐르는 것을 방지하는 정보의 비밀성을 보장하는데는 효과가 있지만 인가등급이 낮은 주체가 보안등급이 높은 객체를 부당하게 훼손하거나 변경할 우려가 있어 정보의 무결성을 보장하지는 못한다.

(표 1) 직접권한과 실제권한의 예

역할이름	직접권한	실제권한
MaxRole	∅	{1,2,3,4,5,6,7,8,9,10,11}
VP1	{9,10}	{1,2,3,4,5,6,7,8,9,10}
VP2	{11}	{1,2,3,4,5,6,7,8,11}
L1	{3,4}	{1,3,4}
L2	{4,5}	{1,2,4,5}
L3	{5,6}	{1,2,5,6}
L4	{7,8}	{2,7,8}
S1	{1}	{1}
S2	{2}	{2}
MinRole	∅	∅

역할기반 접근통제 모델에서 역할 계층의 개념은 많은 이론적 접근통제모델에서 중요한 중심부분으로 역할계층에 의한 역할이 사용자나 주체에 배정되고 역할에 배정된 권한을 정적으로 부여하므로서 수많은 접근권한을 관리하는데 융통성을 제공한다. 역할은 역할계층에서 그 역할의 하위 역할에 할당된 모든 권한들을 상속받으며 특정한 역할에 배정된 사용자는 계층상에서 모든 하위 역할들을 활성화할 수 있다.

그렇지만, 역할계층의 경우 부분순서 지배관계에 따라 상위역할의 경우에 하위역할의 모든 권한을 상속받으므로 권한간의 충돌 및 남용 문제가 발생된다. 그중에서 가장 심각한 것은 상위 역할이 모든 하위 역할에 배정된 권한들에 접근하여 사용할 수 있다는 것이다. 이는 상위 관리자가 여러 하위 직위들의 업무 모두를 충분히 수행할 정도로 유능하거나 경험이 있는 것이 아니기 때문에 조직적으로서 부적절하다. 또한, 역할계층에서 권한의 상속은 항상 상향이기 때문에 역할기반 접근통제모델은 직접적으로 권한이 하향으로 상속되기를 요구하는 정보의 쓰기와 관련된 스타 특성을 지원하지 않는다.

기존의 역할기반 접근통제모델을 사용하여 강제적 접근통제 모델중 BLP모델을 실험한 몇 가지 시도가 있었다[1,11,12].이러한 연구의 동기는 일반적으로 역할기반 접근통제모델이 다양한 접근통제 모델들을 구현하는데 이용될 수 있고, 정책 중립적이라는 것을 증명하는 것이었다. 또한 이들 연구의 기본적인 접근법은 역할계층에서의 상속을 그대로 이용함으로서 권한 남용을 통제할 수 없다는 점을 간과하고 있다.

우리는 이러한 관점에서 다음과 같은 사항에 근거하여 모델을 제안한다.

첫째, 우리는 역할기반접근통제모델에서의 사용자 배정 관계와 강제적 접근통제모델의 주체와 객체의 등급 사이의 명시적 연결을 시도한다.

둘째, 우리는 강제적 접근통제모델에서 접근권한의 획득이 권한상속에 의해서가 아니라 사용자의 보안 등급에 의해서 통제된다는 것이다.

마지막으로 우리는 강제적 접근통제모델에서 스타 특성이 역할기반 모델에서 제약조건을 이용하여 지원하는 임무분리 형태를 제공한다는 것이다.

본 논문에서는 강제적 접근통제모델의 등급과 역할기반 접근통제 모델의 제약조건과 역할계층을 체계적으로 변경함으로서 격자기반 역할그래프 보안 관리 모델을 제안한다. 이 모델에서는 기존의 역할그래프 모델의 역할계층에서 상위역할의 권한남용 문제를 해결하였으며 권한간의 충돌 발생시 제약조건을 통해 주체의 등급을 재조정함으로서 정보의 무결성을 유지할 수 있다. 또한 역할계층에 의한 권한상속 뿐만 아니라 사용자의 보안 등급에 의해서 통제되도록 함으로서 강화된 보안기능을 제공한다. 마지막으로 본 모델을 운영하기 위해 권한충돌을 허용하지 않는 역할그래프 보안 관리 알고리즘을 제시하였다.

3.1 기본적인 정의

다음 정의에서는 기존의 역할기반 접근통제 정책에서 사용되는 집합과 본 논문에서 사용되는 개념을 정형적으로 정의하였다. 본 논문에서는 기존의 역할기반 접근통제 정책에 사용되는 기본 정의를 일반적으로 수용하면서 강제적 접근통제 정책에 적용되는 몇 가지 집합을 새롭게 정의하였는데 이는 주체와 객체의 인가 및 보안등급으로 총 4단계로 구성된다. 또한, 본 논문이 일방향의 정보 흐름을 강조하는 격자에 기반을 둔 모델이므로 역할을 읽기, 추가, 읽고쓰기인 3단계로 일반화하였다. 마지막으로 해당 역할들이 권한을 수행할 때 권한의 접근 모드로는 read, append, write로 나누었다. 이러한 역할과 접근모드는 추후 다양한 역할과 접근모드로 세분화가 가능하리라 본다.

- USERS : 사용자 집합, $\{u_1,...,u_n\}$
- SUBJECTS : 주체의 집합, $\{s_1,...,s_n\}$
- LEVEL : 주체와 객체의 인가 및 보안등급 집합, {c, vi, i, o}
- ROLES : 역할의 집합, {xR, xA, XRW}
- ACCESS_MODE : 접근모드의 집합, {read, append, write}
- OBJS : 객체의 집합 , $\{obj_1,...,obj_n\}$
- PRMS : $2^{(ACCESS\_MODE \times OBJS)}$, 권한의 집합, $\{p_1,...,p_n\}$
- PA $\subseteq$ PERMS$\times$ ROLES = OPS$\times$ OBJS$\times$ ROLES 다대다 권한 대 역할배정 관계
- UA $\subseteq$ USERS$\times$ ROLES , 다대다 사용자 대 역할 배정 관계
- RH $\subseteq$ ROLES$\times$ ROLES , 부분순서쌍 역할계층
- assigned_users :(r:ROLES) $\rightarrow$ 2^{USERS}, 역할 r에 배정된 사용자 집합으로 다음과 같이 정형적으로 기술된다. assigned_users(r) = {u $\in$ USERS | (u,r) $\in$ UA}
- assigned_permissions(r:ROLES) $\rightarrow$ 2^{PRMS} 역할 r에 배정된 권한 집합으로 다음과 같이 정형적으로 기술된다. assigned_permissions(r) = {u $\in$USERS | (p,r) $\in$ PA}
- subject_user(s:SUBJECTS) $\rightarrow$ USERS ,주체와 연관된 사용자 집합으로 다음과 같이 정형적으로 기술된다. subject_user(si) $\subseteq$ {u $\in$ USERS | (s_i,r) $\in$ UA}
- subject_roles(s:SUBJECTS) $\rightarrow$ 2^{ROLES}, 주체와 연관된 역할 집합으로 다음과 같이 정형적으로 기술된다. subject_roles(s_i) $\subseteq$ {r $\in$ ROLES | (subject_user(s_i),r) $\in$ UA}

본 논문에서 제안한 (그림 5)의 격자기반 역할그래프 보안그래프 관리 모델에서 적용되는 개념

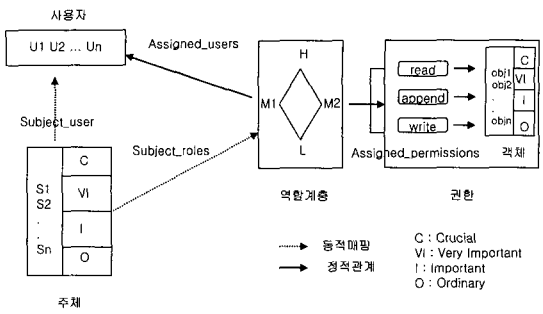
인 LEVEL과 ROLES, 그리고 ACCESS_MODE중, LEVEL은 주체와 객체의 인가 및 보안등급인 c(crucial), vi(very important), i(important), o(ordinary)로 구성되며 이들의 관계는 c > vi > i > o 를 갖으며, ACCESS_MODE는 일반적인 접근모드인 read, append, write로 구성된다. 또한 격자에 기반을 둔 역할그래프 구조이므로 일반적인 접근모드에 관련된 역할로서 xR(읽기역할), xA(추가역할), xRW(읽고쓰기역할)로 역할을 구분하였다. 여기에서 x는 객체에 배정된 보안등급을 의미한다.

우리의 모델에서 권한은 시스템 특성에 따라 의존적인데, 사용자는 많은 시스템 계정을 소유할 수 있으며 다양한 시스템을 통하여 자원에 접근할 수 있다. 사용자들은 다음 역할집합중 하나인 xR, xA, xRW에 배정된 사용자에 포함되어야 한다. 예를 들어 사용자 u_i가 xR의 역할을 수행하기 위해서는 정적관계에서 $u_i \in$ assigned_users(xR)과 $p_i(obj_i, read) \in$ assigned_permissions(xR)이 만족되어야한다. 또한, 동적매핑관계로서 해당 시스템 계정 중 특정 세션, 즉 주체에 사용자 u_i가 포함되는 $u_i \in$ subject_user(s_i)과 해당 주체가 수행 가능한 역할에 포함되어야하는 $r_i \in$ subject_roles(s_i)가 만족되어야 한다. 특히, 역할을 수행하는 주체 s_i들과 객체 obj_i들은 정보의 무결성을 보장하기 위해 인가 및 보안등급을 비교하여 제약조건에 따라 접근여부가 결정된다.

여기에서 접근통제 모델 관리 과정의 사용자 배정단계에서 사용자에 대한 역할의 권한부여가 결정되는 정적과정과 사용자가 실제 역할을 사용하는 세션 수행시 주체의 역할의 권한부여가 결정되는 동적매핑과정으로 나뉜다.

정적관계는 매우 엄격한 무결성 유지기능을 제공하는 반면 각 역할에 많은 수의 사용자가 필요하게 되어 시스템 운용에 따른 부담이 증가하고 시스템 운용이 유연하지 못한 단점을 가진다.

동적매핑은 실제 사용자가 수행하는 세션의 주체가 실행하는 실행시점에서 주체와 객체 및 역할간의 등급비교 및 제약조건에 따라 권한부여가



(그림 5) 격자기반 역할그래프 보안 관리 모델

결정되므로 시스템의 구성과 운용의 유연성을 보장하게 된다.

3.2 권한집합과 제약조건

3.2.1 권한집합

우리는 역할의 부분순서관계인 < R, ≤ >가 존재할 때, 만일 (u, r) ∈ UA이면 우리는 사용자 u는 역할 r에 명시적으로 배정되어 있다고 말할 수 있고, 또한 사용자 u는 r의 모든 하위역할에 배정되었다고 말할 수 있다. 우리는 사용자 u에 명시적으로 배정된 역할들의 집합을 R(u)로 표기한다. 시스템과 사용자의 상호작용은 세션에 의해서 이루어지는데, 하나의 세션은 사용자 u에게 배정된 역할들의 부분집합을 활성화한다. 만일 (p,r) ∈ PA이면 우리는 권한 p는 역할 r에 명시적으로 배정되어 있다고 말할 수 있다. 우리는 권한 p에 명시적으로 배정된 역할들의 집합을 R(p)로 표기한다.

우리는 권한집합 P를 다음과 같이 세 가지 유형으로 나눈다. 권한집합 중에서 읽기와 관련된 권한집합을 P_R, 추가와 관련된 권한집합을 P_a, 읽기쓰기와 관련된 권한집합을 P_{RW}로 구분한다.

각 권한 p∈P는 p의 실제 권한으로 구성된 역할을 $R_E(p)$라고 했을 때 우리는 함수 R_E를 아래와 같이 정의한다.

$$\triangleright\ R_E(p) = \{\ \downarrow R(p),\ \text{if } p \in P_R \mid \uparrow R(p),\ \text{if } p \in P_a \mid \updownarrow R(p),\ \text{if } p \in P_{RW}\ \}$$

만약 p가 P_R에 속한다면 무결성 측면에서 read up 정책에 따라 권한 p를 수행하는 역할집합은 역할계층 아래에서 많은 권한을 갖는다($\downarrow R(p)$).

만약 p가 P_a에 속한다면 무결성 측면에서 write down 정책에 따라 권한 p를 수행하는 역할집합은 역할계층 위쪽에서 많은 권한을 갖는다($\uparrow R(p)$).

만약 p가 P_{RW}에 속한다면 무결성 측면에서 read up과 write down 정책에 따라 read 측면의 상향과 write 측면의 하향이 공유하는 역할계층의 권한을 갖는다.($\updownarrow R(p)$).

표준 역할기반 접근통제 모델에서 만일 $p \in P_a$ 이고 $(p, r) \in PA$ 라면 p는 $\uparrow r$안의 모든 역할들에 할당된다. 그러므로 r보다 상위에 있는 역할 중에 적어도 하나의 역할에 배정된 임의의 사용자는 권한 p를 실행할 수 있다.

3.2.2 제약조건

본 모델을 지원하기 위해 우리는 주체와 객체의 등급에 영향을 미치는 적절한 제약조건을 강화할 필요가 있다. 본 모델에서 각 사용자는 유일한 인가등급을 가지며 각 객체는 보안등급을 갖는다.

<사용자배정 제약조건>

각 사용자는 사용자 배정 상에서 정확하게 세 개의 역할 xR과 xA, xRW에 배정된다. 단, x는 객체에 의해 배정된 등급을 의미한다. 각 역할에 할당된 객체들의 보안등급을 x-level(R)이라 할 때,

xR의 경우 주체의 인가등급이 역할의 보안등급인 r-level(R)에 지배되어야 하며, 이를 정형적으로 기술하면 다음과 같다.

▷ assigned_users(xR) : r-level(R) dominates level(s_i) ;

xA의 경우 주체의 인가등급이 역할의 보안등급인 a-lelvel(R)을 지배해야하며 이를 정형적으로 기술하면 다음과 같다.

▷ assigned_users(xA) : level(s_i) dominates a-level(R) ;

xRW의 경우 r-level(R)이 a-level(R)을 지배하면서 주체의 인가등급이 r-level(R)과 a-level(R)사이의 등급을 가져야 하며 이를 정형적으로 기술하면 다음과 같다.

▷ assigned_users(xRW) : r-level(R) dominates a-level and a-lvel(R) $\leq$ level(s_i) $\leq$ r-level(R) ;

<권한배정 제약조건>

만약 (obj, m)을 xR, xA, xRW에 배정 시 권한 간의 충돌 발생 시 주체 등급을 재조정한다.

최소상한 : xR의 경우 (obj, m)의 집합인 권한 내에서 객체 간에 충돌이 발생한 경우 주체의 인가등급을 객체의 최소상한등급으로 재조정하며 이를 정형적으로 기술하면 다음과 같다.

▷ assigned_permissions(xR) : if (effective_privilege(xR) $\in$ P-conflicts) then level(s_i) $\leftarrow$ GLB(level(obj_i))

최대하한 :xA와 xRW의 경우 (obj, m)의 집합인 권한내에서 객체간에 충돌이 발생한 경우 주체의 인가등급을 객체의 최대하한등급으로 재조정하며 이를 정형적으로 기술하면 다음과 같다.

▷ assigned_permissions(xA or xRW) : if (effective_privilege(xA or xRW) $\in$ P-conflicts) then level(s_i) $\leftarrow$ LUB(level(obj_i))

이러한 주체의 등급을 재조정함으로서 기존의 정책인 정보의 무결성을 해치지 않으면서 권한간의 충돌을 조정하여 운영할 수 있다.

<세션 제약조건>

$S(u) \subseteq \downarrow R(u)$인 세션이 주어졌을 때, 사용자 u가 권한 p를 실행하고자 하는 요청은 사용자 u

가 권한 p의 실제 권한 중에 하나를 활성화시킬 수 있을 때만 승인된다.

$\triangleright$ $S(u) \cap R_E(p) \neq \varnothing$

단, S(u)는 사용자 u가 실행하는 세션이며, $\downarrow$ R(u)은 사용자 u가 배정된 역할들의 하위역할 그리고 $R_E(p)$는 실제권한을 의미한다.

3.3 알고리즘

격자기반 역할그래프 보안 관리 모델은 역할 및 권한을 추가하거나 삭제하는 역할 관리 기능을 수행할 때 (표 2)에서처럼 새로 생성할 역할의 상위 역할 및 하위 역할의 등급비교를 수행한 후 생성 및 삭제된다. 이러한 비교 절차는 정보의 무결성을 보장하기 위한 read up, write down 정책을 유지하면서 역할을 생성하기 위함이다.

역할 추가 알고리즘의 입력값은 역할그래프, 역할명과 직접권한들의 집합으로 표현되는 새로 추가할 역할, 그리고 인접한 상위, 하위 역할 그리고 권한충돌의 집합으로 구성된다. 알고리즘에 의해 생성되는 출력값은 새로운 역할이 추가된 역할그래프로 기존의 역할그래프의 특성은 유지되어야 한다.

새로운 역할이 추가되기 위해서는 새로운 역할과 상위역할 그리고 하위역할간의 에지 생성시 사이클이 생성되어서는 안된다. 또한, 임의의 역할 r_i, r_j에 대하여 이들의 실제권한이 같은 경우 중복 역할로 간주되어 취소되고, 최대역할을 제외한 역할의 실제권한이 권한충돌집합에 포함되는 경우 취소되어야 한다.

(표 2) 역할그래프 보안 관리 알고리즘

```
BEGIN
 IF level(xR) ≥ senior(xR) and level(xR) ≤ junior(xR)
THEN  /*read up 정책 */
  IF level(xA) ≤ senior(xA) and level(xA) ≥ junior(xA)
THEN /*write down 정책 */
 roleAddtiton(); /*역할추가알고리즘*/
 roleDeletion(); /*역할삭제알고리즘*/
 PrivilegeAddition(); /*권한추가알고리즘*/
 PrivilegeDeletion(); /*권한삭제알고리즘*/
 ENDIF
ENDIF
END
```

역할추가알고리즘 :RoleAddition(RG, xn, xSeniors, xJuniors, P-Conflicts)

Input : RG = < xR, → > /*등급 x를 갖는 역할그래프*/

xn : 추가될 새로운 역할로서 (rname, rpset)으로 구성(단, x는 등급을 의미)

xSeniors : n의 상위 역할

xJuniors : n의 하위 역할

P-Conflicts : 권한 충돌의 집합

output : 새로운 역할이 추가된 역할그래프로 역할그래프의 특성은 유지

method:

Var xr, xr_i, xr_j, xr_s : role;

Begin

xR : xR ∪ {xn}; /*RG에 새로운 노드가 추가*/

For all xr_s ∈ xSeniors DO add the edge xn → xr_s;

For all xr_j ∈ xJuniors DO add the edge xr_j → xn;

If RG has cycles then abort;

For all xr_i, xr_j ∈ xR Do

If 실제권한(xr_i) = 실제권한(xr_j) then abort;

For every role xr ∈ {xn} ∪ xR - MaxRole Do

If 실제권한(xr) ∈ P-Conflicts then abort;

end.

역할삭제 알고리즘의 입력값은 역할그래프와 삭제될 역할 그리고 권한충돌집합으로 구성된다. 역할그래프에서 삭제될 역할에 인접한 상위역할과 하위역할을 찾고 하위역할에서 상위역할로 가는 에지를 생성한다. 그리고 삭제할 역할에서 상위역할로 가는 에지와 하위역할에서 삭제될 역할로 오는 에지를 차례로 제거고 역할집합에서 삭제할 역할을 제거한다.

역할삭제 알고리즘 : RoleDeletion(RG, xn, P-Conflicts)

Input : RG = < xR, → > /*역할그래프*/
xn : 삭제할 역할
P-Conflicts : 권한 충돌의 집합
Output : 역할 xn이 삭제된 역할그래프로 역할그래프의 특성은 유지
Method :
Var xS, xJ : 역할 집합;
xr_j, xr_s : 역할;
xS : FindImmediateSeniors(RG, xn); /*삭제할 역할의 인접한 상위 역할*/
xJ : FindImmediateJuniors(RG, xn); /*삭제할 역할의 인접한 하위 역할*/
For all $xr_j \in xJ$ Do
For all $xr_s \in xS$ Do
add edge $xr_j \rightarrow xr_s$;
For all $xr_s \in xS$ Do
remove edge $xn \rightarrow xr_s$;

For all $xr_j \in xJ$ Do
remove edge $xr_j \rightarrow xn$;
remove xn from xR;

권한추가 알고리즘은 역할그래프, 권한이 추가되어질 역할, 역학 r에 추가되어질 권한, 그리고 권한충돌집합이 입력값으로 주어진다. 새로 추가될 권한이 기존의 역할 n에 존재하면 추가될 필요가 없으며 그렇지 않다면 직접권한과 실제권한에 추가된다. 그러나 추가되어진 역할그래프가 사이클을 형성하거나 권한충돌집합에 포함될 경우 취소되어진다.

권한추가알고리즘 : PrivilegeAddition(RG, xn, p, P-Conflicts)

Input : RG = < xR, → > /*역할그래프*/
xn : 권한이 추가되어질 역할
p : 역할 xr에 추가되어질 권한
P-Conflicts : 권한 충돌의 집합
Output : 역할 xr에 p가 추가된 역할그래프로 역할그래프의 특성은 유지
Method :
Var xr :역할;
Begin
If p ∈ 실제권한(xn) /*새로 추가될 권한이 기존의 역할 xn에 존재하면
then return;
else
직접권한(xn) = 직접권한(xn) ∪ {p};
실제권한(xn) = 실제권한(xn) ∪ {p};
If RG has any cycles then abort;
For every role xr ∈ {xn} ∪ xR - MaxRole Do
If 실제권한(xr) ∈ P-Conflicts then abort;
end

권한삭제 알고리즘은 역할그래프, 삭제되어질 권한을 포함하는 역할, 역할 n으로부터 삭제되어질 권한 그리고 권한충돌 집합으로 구성된다. 만약 삭제할 권한이 직접권한집합에 존재하지 않는다면 상관없지만 그렇지 않다면 직접권한집합과 실제권한집합에서 제거되어야 한다.

권한삭제알고리즘 : PrivilegeDeletion(RG, xn, p,P-Conflicts)

Input : RG = < xR, → > /*역할그래프*/
xn : 삭제되어질 권한을 포함하는 역할
p : 역할 n으로부터 삭제되어질 권한
P-Conflicts : 권한 충돌의 집합
Output : 역할 xn에 p가 삭제된 역할그래프로 역할그래프의 특성은 유지
Method :
Var xr :역할;
Begin
If p ∉ 직접권한(xn) then return; /*새로 추가될 권한이 기존의 역할 xn에 존재하면
else

```
직접권한(xn) = 직접권한 - {p};
실제권한(xn) = 실제권한 - {p};
  For all xr ∈ xR Do
    If 실제권한(xr) = 실제권한(xn) then abort;
end
```

3.4 비교분석

본 논문에서 제시한 모델은 기존의 역할그래프 모델과 비교해 다음과 같은 기능들을 제공한다.

첫째, 기존의 역할그래프에서 Max_Role의 경우 하위의 모든 권한을 상속받으므로 인해 해당 역할의 모든 권한을 수행할 수 있다. 이는 만약 Max_Role을 수행하는 사용자가 부당하게 권한을 수행한다고 한다면 이를 견제할 수 있는 장치가 전혀 없게 된다. 본 논문에서는 이를 주체와 객체의 등급과 역할계층구조에 따라 역할배정을 조정함으로서 이를 조정할 수 있다.

둘째, 기존의 역할그래프에서는 권한 충돌이 발생할 경우 권한간의 충돌쌍들의 집합을 정의해 놓고 Max_Role을 제외한 어떠한 역할도 상충되는 권한들을 동시에 가질 수 없도록 제한하고 있다. 하지만 이러한 충돌 처리 방식은 현실에 비추어볼 때 너무 엄격하고 제한적이다. 그러므로 권한 삽입시 충돌이 발생하는 경우 기존 권한에서 문제가 되는 권한의 등급조정을 통해 해결할 수 있다.

셋째, 우리는 강제적 접근통제모델에서 접근권한의 획득이 권한상속 뿐만 아니라 사용자의 보안레벨에 의해서 통제되도록 함으로서 역할에 의한 임무분리 특성과 주체나 객체의 보안특성에 따라 접근을 통제함으로서 강화된 보안기능을 제공한다.

4. 결 론

정보생산량의 증가로 인하여 데이터베이스의 규모가 증가하고 이에 따른 사용자별 정보 접근의 요구사항 또한 다양해져서 이들로부터 정보를 효율적으로 저장 및 관리하기 위한 보안이 필수적인 요소가 되었다. 이에 따라 허가 받은 사용자나 주체만이 자원에 접근할 수 있도록 통제하는 접근통제 기법이 시스템 관리자나 프로그램 개발자에게는 무엇보다 중요한 사항이다.

본 논문의 특성은 적합한 등급을 갖는 사용자가 데이터베이스에 접근할 때 부당한 자료의 유출과 변경이 이루어지지 않으면서 그 사용자에게 허가된 데이터베이스를 효과적으로 사용할 수 있도록 하는데 있다. 이러한 목적을 달성하기 위해 강제적 접근통제와 역할기반 접근통제 그래프를 이용해 데이터의 무결성과 보안을 유지하면서 역할을 효율적으로 관리하고 제어할 수 있는 보안 모델을 제시하였다. 기존의 역할그래프에서 상위에 있는 MaxRole의 경우 하위 역할의 모든 역할을 상속받으므로서, 부당한 목적을 가지고 사용될 경우 정보의 기밀성과 무결성에 문제가 발생할 수 있다. 따라서 사용자 및 역할 배정시 정보의 무결성을 유지하기 위해서는 제약조건에 의해 자원의 접근이 선택적으로 제한되어야 한다. 또한 역할내의 권한간의 충돌이 발생할 경우 정보의 가용성을 침해하지 않으면서 일관된 접근통제를 제공하기 위해 역할에 의한 임무분리 특성과 주체나 객체의 보안특성에 따라 보안등급을 조정함으로서 충돌문제를 해결하고 보안을 강화할 수 있다.

정보 통신망의 관리 정보를 이용하는 사용자의 환경이 동적으로 변화하는 현대의 네트워크 환경에서 다양한 접근통제 정책들 간의 연관성의 연구가 필요하다고 볼 때, 본 논문은 실용적인 정보의 이용성과 무결성을 동시에 보장하기 위한 접근통제 정책들의 상호연관성 연구에 기여하리라 본다.

참 고 문 헌

[1] S. Osborn, R. Sandhu, Q. Munawer, "Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies," ACM Transactions on Information and System Security, Vol.3, No.2, pp.85-106, 2000.

[2] 이금순, 김영호, 원용관, "상세 접근 제어를 위한 데이터베이스 보안 모델," 정보처리학회 추계학술대회 논문집, 제9권 제2호, 2002.

[3] R. Sandhu, V. Bhamidipati, Q. Munawer, "The ARBAC97 model for role-based administration of roles," ACM Transactions on Information and System Security, Vol1, No.2, pp.105-135, 1999.

[4] S. Osborn, "Mandatory access control and role-based access control revisited," In Proceeding of the 2nd ACM Workshop on RBAC, pp.31-40, 1997.

[5] E. Bertino, P.A. Bonatti, "TRABAC:A Temporal Role-Based Access Control Model," ACM Transactions on Information and System Security, Vol.4, No.3, pp.191-223, 2001.

[6] J. Wang, S. Osborn, "A Role-Based Approach to Access Control for XML Databases," ACM Symposium on a Access Control Models & Technologies, pp.70-77, 2004.

[7] J. Cramption, "Specifying and Enforcing Constraints in Role-Based Access Control," ACM Symposium on a Access Control Models & Technologies, pp.43-50, 2003.

[8] 문창주, 박대하, 박성진, 백두권, "임무분리와 역할 계층구조를 고려한 대칭 RBAC 모델," 정보과학회 논문지, 제30권 제12호, pp.699-707, 2003.

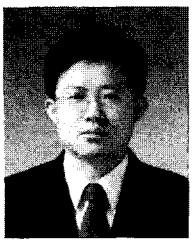
[9] G. Neumann, M. strembeck, "An Approach to Engineer and Enforce Context Constraints in an RBAC Environment," ACM Symposium on a Access Control Models & Technologies, pp.65-79, 2003.

[10] 정유나, 황인준, "권한세분화를 이용한 역할그래프 모델에서의 유동적 권한삽입 연산," 정보과학회 추계학술대회 논문집, 제30권, 제2-1호, pp.637-639, 2003.

[11] KUHN, D. "Role based access control on MLS systemss without kernel changes," In Proceedings of Third ACM Workshop on Role-Based Access Control, pp.25-35, 1998.

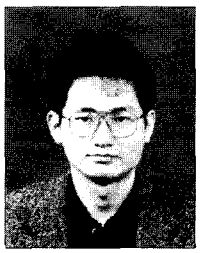
[12] SHANDHU, R. "Role hierarchies and constraints for lattice-based access controls," In Proceedings of Fourth European Symposium on Research in Computer Security, pp.65-79, 1996.

◐ 저 자 소 개 ◑



최 은 복

1992년 전남대학교 전산학과 졸업(이학사)
1996년 전남대학교 대학원 전산학과 졸업(이학석사)
2000년 전남대학교 전산학과 졸업(이학박사)
2001년 순천제일대학 인터넷정보학부 전임강사
2002년- 현재 전주대학교 정보기술공학부 조교수
관심분야 : 통신망관리, 네트워크보안 etc.
E-mail : ebchoi@jj.ac.kr



박 주 기

1986년 전남대학교 전산학과 졸업(이학사)
1993년 전남대학교 대학원 전산학과 졸업(이학석사)
1993~현재 KT 책임연구원
관심분야 : 인터넷 보안, 인터넷트래픽 분석 및 모델링 etc.
E-mail : jugipark@kt.co.kr

김 재 훈

1994년 전주대학교 통계학과 졸업(학사)
1996년 전주대학교 대학원 통계학과 졸업(석사)
2005년 전주대학교 대학원 통계학과 졸업(박사)
2005~현재 전주대학교 교양학부 객원교수
관심분야 : 신경망, 시계열, etc.
E-mail : muggeby@jj.ac.kr