제25회 한국정보처리학회 춘계학술발표대회 논문집 제13권 제1호 (2006. 5)

RBAC 을 위한 역할 정보 저장소의 설계

김원일, 하홍준, 이창훈
건국대학교 컴퓨터공학과
e-mail : {unangel, greatsk, chlee}@konkuk.ac.kr

Design of Role Information Storage for Role-Based Access Control

Won-iL Kim, Hong-Joon Ha, Chang-Hun Lee
Dept. of Computer Engineering, Konkuk University

요 약

RBAC 은 기업이나 조직이 필요로 하는 유연한 조직의 관리와 각 업무가 갖는 책임과 권한을 정교하게 제공할 수 있는 접근제어 방법이다. RBAC 은 사용자와 역할 그리고 허가로 구성되어 사용자와 역할간의 관계, 역할과 허가의 관계를 통해 접근제어를 수행하며 역할의 범위에 따라 유연한 접근제어를 제공한다. 여기에 정교한 접근제어를 제공하기 위해 각 역할이 수행할 수 있는 작업의 한계를 제약조건으로 두어 제약조건에 위배되는 작업을 수행할 수 없도록 구성되어 있다. 그러나, 하나의 역할 정보만을 유지하는 형태로 구성되어 있어 의무 분리나 상속에 의한 여러 가지 연산 부하가 발생한다. 본 논문에서는 기본적인 RBAC 의 기능을 만족하고, 최근 발표된 NIST 의 RBAC 표준을 준수하며, 허가 및 역할 변경에서 발생할 수 있는 여러 부하를 줄일 수 있는 역할 저장소의 모델을 제안한다.

1. 서론

접근제어는 정보시스템의 자원을 사용하기 위해 정당한 사용자임을 증명하는 인증과정을 거쳐 로그인한 사용자가 허가된 범위 내에서 시스템 자원과 정보에 대한 접근을 허용하는 기술적인 방법으로, 접근의 허용과 거부는 허가에 의해 결정된다. 이렇게 허가에 의한 접근 제어는 임의적 접근제어, Discretionary Access Control 과 강제적 접근제어, Mandatory Access Control 의 형태가 가장 많이 사용되었다[1]. 임의적 접근제어는 사용자가 자원(객체)에 접근하고자 할 때, 접근하는 주체가 해당 자원에 대한 허가권의 존재여부를 확인하고 허가권을 부여하는 접근제어 방법이다. 강제적 접근제어는 자원에 접근하는 주체가 갖는 보안 레이블과 해당 자원이 갖는 보안 레이블 정보를 비교하여 정책에 합당한 경우에 허가권을 부여하는 방법이다[1, 3].

임의적/강제적 접근제어는 접근 주체와 자원간의 관계 설정으로만 접근의 허용과 거부를 결정하므로, 기업이나 조직과 같이 주체 정보의 변경이 자주 발생하는 경우에 많은 부하가 발생한다. 즉, 주체 정보의 변경이 자원과 허가에 미치는 영향으로 해당 자원에 대한 허가 정보가 지속적으로 변화해야 하며, 지속적인 변화 속에서 허가가 정상적으로 이루어지는지를 검증해야 하는 문제점이 있다. 따라서 주체와 자원이 지속적으로 변화하는 조직(정보 시스템)에 적합한 접근제어는 여러 변화에 유연하게 대처할 수 있어야 하며, 대처한 내용이 정교함을 내포해야 한다[2, 5].

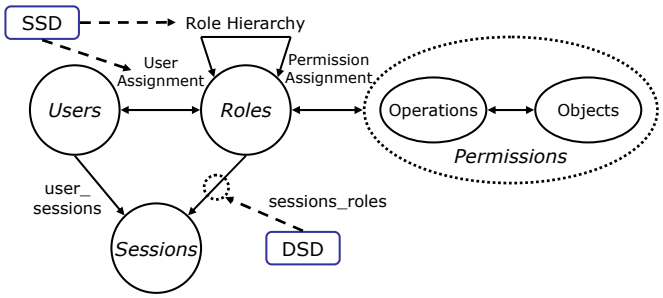
RBAC 은 이러한 관리와 검증의 문제점들을 해결하기 위해 고안된 접근제어로, 전통적인 접근제어에서 발생하는 문제점들을 역할이라는 요소의 도입으로 해결한다. 역할 요소 외에 의무 분리와 상속을 통해 전통적인 접근제어가 내재하고 있는 문제점에 대해 유연하게 대처할 수 있다. 그러나, 최근 발표한 RBAC 표준을 비롯하여 발표된 RBAC 모델들은 내부 동작에 필요한 역할의 저장 구조에 대한 모델이 없어 구현

시에 필요에 의해 저장된 정보 활용과 처리 구조가 변화하고 있다. 또한 핵심 요소인 역할은 중복될 수 있으나 사용시에는 하나의 역할만 적용된다는 특성 때문에 저장 구조에 대한 연구는 거의 없었다. 본 논문에서는 역할의 특성을 만족하면서도 여러 환경에서 사용이 가능하고, 표준 RBAC 모델을 지원할 수 있는 역할 정보 저장소 모델을 제안한다. 제안한 저장소 모델은 실행시간에만 적용할 수 있는 응용 프로그램 모델로 그 한계를 정한다.

2 장에서는 RBAC 이 갖는 특징과 핵심 구성 요소에 대해 알아보고, 3 장에서는 RBAC 이 갖는 핵심 구성 요소들의 조건을 만족하면서 유연하고 정교한 접근제어를 제공하기 위한 역할 저장소와 지원 방법을 제안한다. 5 장에서는 결론과 앞으로의 연구방향을 제시한다.

2. RBAC 의 특징

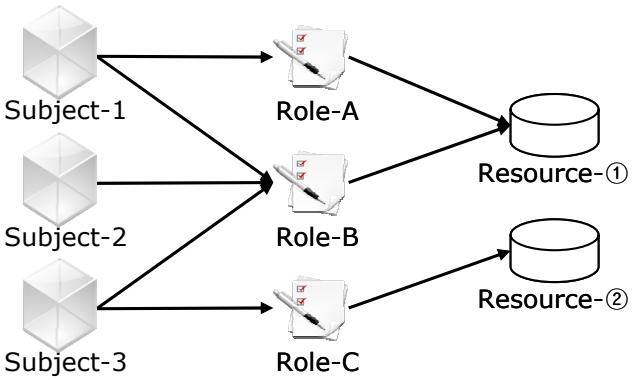
전통적 접근제어인 임의적/강제적 접근제어가 기업과 같은 조직 환경에 적합하지 않음을 인식한 NIST 가 다양한 응용환경에 적용 가능한 새로운 접근제어의 필요성으로 접근제어 연구를 시작하였다[2]. 새로운 접근제어의 연구결과로 주체와 자원 이외의 도입된 추상적 개념이 역할, Role 이다(RBAC0)[6]. 다음으로 각 역할은 조직이 갖는 중복 업무의 특성에 맞게 상속 기능을 추가한 모델이 발전하였다(RBAC1). 상속 기능과는 다른 방향으로, 역할 중복으로 인해 최소 특권의 원리가 유지될 수 없다는 문제점을 해결하기 위해 각 요소에 제약을 가하여 각 주체가 필요 이상의 특권을 가질 수 없도록 의무분리 제약을 추가한 모델도 발전하였다(RBAC2). 그리고 이렇게 분리된 상속과 제약을 하나로 합하여 만든 모델(RBAC3)이 대두되었고[], 여기에 NIST 가 정적 의무분리(MC0)를 추가하고, 동적인 의무분리(MC1)까지 추가하여 현재 RBAC 이 갖는 대부분의 특징을 갖게 되었다. <그림 1>



<그림 1> RBAC 의 구조[4]

1) 역할(Role)

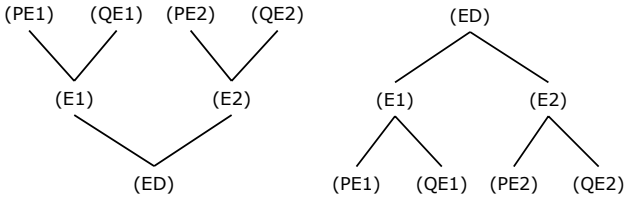
역할은 주체들과 연결되어 있으면서 또한 자원과도 연결되어 있는 추상적 개념 요소이다. 역할은 주체와 자원에 각각 다대다의 관계를 갖는다(<그림 2>). 따라서 주체가 동일한 자원에 접근할 때, 적용할 역할에 맞는 허가에 따라 접근의 허용과 거부가 결정된다. 역할은 사용자, 허가와 각각의 관계를 가지고 독립적으로 존재하며, 필요에 따라 사용자에게 할당되거나 제거된다. 즉, 담당 업무자 정보의 추가, 변경 그리고 삭제 시에 역할의 할당과 해제를 통해 사용자의 접근 제어 정보 변경 및 관리를 마칠 수 있고, 모든 변경 사항은 역할에 연결된 사용자와 자원을 변경하는 것으로 일괄 적용이 가능하다. 이러한 역할의 특징 때문에 임의적 접근제어나 강제적 접근제어에 비해 높은 수준의 유연성과 정교한 접근제어가 가능하다.[7]



<그림 2> 주체와 역할 및 자원의 관계

2) 상속(Hierarchy)

상속은 기업과 조직의 특성에서 발생되는 동일 업무의 중복에 의해 추가된 개념으로, 하나의 자원에 대해 접근하는 하나의 부서가 모두 동일한 권한을 가지도록 구성하는 구성 요소이다. 즉, 직책에 따른 역할을 수행하면서 특정 자원에 대한 접근을 부서 전체가 가능해야 할 때 적용된다. 이것은 역할이 주체에 대해 다대다의 관계를 가지기 때문에 가능하고, 상속은 대부분 트리 계층으로 표현되며 조직의 특징에 따라서 역트리 계층으로 표현되기도 한다(<그림 3>). 트리 계층은 자원의 통합에 강하나 자원 공유에는 약하며, 반면에 역트리 계층은 자원의 공유에 강하나 자원 통합에는 약하다는 서로 다른 특징이 있다. 필요에 따라서 자원의 공유와 통합을 위해 혼합하여 사용할 수 있다[1, 2, 7].



<그림 3> 트리 계층과 역트리 계층

3) 의무분리(Separation of Duty)

의무 분리는 주체와 역할이 자원에 접근할 때, 주체와 역할이 갖는 다대다 관계로 인해 가져서는 안 되는 권한이 주체에게 할당되는 것을 막는 구성 요소이다. 즉, 주체가 여러 역할을 할당 받는 도중 상호간에 침범하지 말아야 하는 권한이 역할간에 암시적으로 허용되는 경우나, 상위 역할의 권한을 이용하여 하위 역할의 권한을 상속받는 명시적인 권한의 허용 범위를 벗어나는 행위를 사전에 탐지하고 방지하는 기

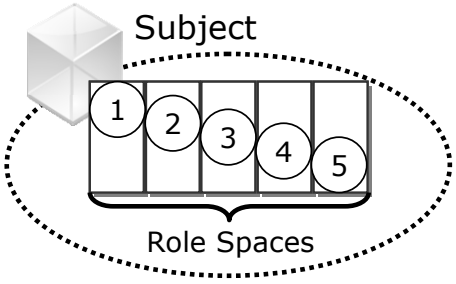
능이다. 이러한 권한의 감시는 임의적 접근제어나 강제적 접근제어보다 더욱 정교한 접근 제어 관리가 가능하도록 한다.

3. 제안 모델

RBAC 은 기술 자체가 갖는 다양성과 유연성 때문에 표준 제정의 어려움이 많았으나, 2006 년 초에 새로운 표준 참조 모델을 NIST 에서 발표[3]하여 RBAC 발전에 새로운 초석을 마련하였다. 따라서 제안된 모델은 표준 참조 모델을 충족하면서도, 여러 환경에서 사용할 수 있도록 구성하는데 초점을 맞추었다.

1) 저장소 모델

먼저 주체인 사용자가 정당한 사용자임을 증명하는 인증과정을 거쳐 로그인하면 <그림 4>와 같은 형태의 Role Spaces 를 갖게 된다. 각 Role Spaces 는 하나의 역할이 포함되는 영역으로, 각 주체가 가질 수 있는 역할의 집합 또는 역할 저장 슬롯이라고 할 수 있다. 이 슬롯에 사용자가 수행할 수 있는 역할을 할당 받고, 할당 받은 역할은 상속이 가능하다. 즉, 개인의 고유한 역할을 비롯하여 상속에 의한 역할 및 조직에서 갖는 역할 등을 포괄하여 저장할 수 있는 공간이다. 기본적으로 공간은 5 개의 공간이 할당되며 이 공간은 Role-Set 에서 해당 사용자가 갖는 Role 을 저장한다. Role Spaces 를 5 개로 설정한 것은 기본적으로 조직에서 사용자가 갖는 역할이 부서 구성원, 팀원 및 개인 고유의 역할로 추상화할 때, 최대 3 개를 넘지 않는다는 것에 기초하여 설정하였다. 조직 구성원 이외의 개인이 갖는 역할을 포함하고, 필요 시에 임시적으로 할당할 수 있는 임시적 역할을 고려하여 총 5 개로 구성하였다.



<그림 4> Role 과 Role Spaces

각 Role Spaces 는 <그림 4>와 같이 고유한 번호를 갖는데 번호의 구성은 다음과 같다. 1 번부터 4 번까지는 부서에서 상속 받은 역할, 직급에 따른 고유 역할 등의 역할을 할당 받을 수 있는 슬롯이고, 5 번은 사용자 고유의 개인 역할(Private Role)이 할당된다. 또한 각 역할은 한번에 하나의 역할만이 활성화 되어야 한다는 고유한 특성을 만족하기 위해 현재 활성화된 역할공간 번호를 표시하는 플래그를 이용한다.

제안된 저장소는 동적으로 할당되는 역할이 아니라 정적으로 할당된 역할의 적용만을 다룬다. RBAC 이 동적 의무 분리를 적용해야 하는 이유는, 사용자와 역할이 각각 다대다 관계가 요구되기 때문이다. 서로 연관성이 없는 2 개 이상의 역할이 충돌을 일으키는 역할이고, 이 역할을 사용자의 특성에 의해 모두 접근할 수 있다고 할 때, 발생되는 동적인 역할의 할당 연관관계가 필요하므로 연산 부하가 발생하게 된다. 저장소는 사용자가 갖는 역할을 정적으로 저장하고 있으므로, 다대다 관계에 의한 연산 부하를 줄일 수 있다.

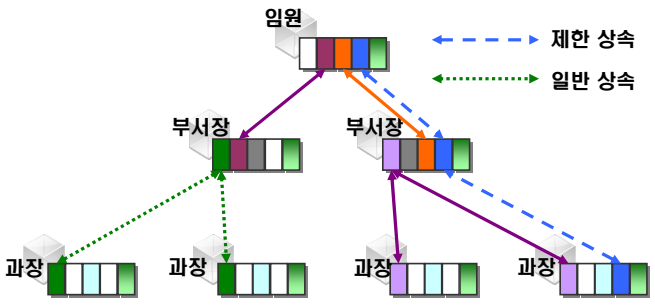
NIST 의 표준 참조 모델의 기본이 되는 Flat RBAC 은 사용자-역할, 허가-역할 간의 할당을 다대다의 관계를 가지도록 요구한다[3]. Role Spaces 는 다대다의 관계를 가지므로 RBAC 의 본질적인 측면을 만족한다.

2) 저장소 사용

2-1 상속

제안된 저장소 모델에서는 상속을 Role Space 번호를 이용하여 수행한다. 고유한 번호를 이용하여 역할을 상속하면 사용자 슬롯의 할당 여부를 통해 역할의 할당 여부를 알 수 있다. 그리고 각 공간이 고유한 번호를 가지기 때문에 상속을 통한 역할의 할당은 해당 슬롯이 비어 있어야 한다는 제약이 따르게 된다. 관리자는 이 제약을 이용하여 전체 사용자의 역할 할당에 대한 정책을 세우거나 조절할 수 있다. 예를 들어, 관리부에 속한 직원들이 Role Space 1 번에 부서 역할을 할당 받도록 정책을 세우면, 다른 역할을 할당할 때 해당 Role Space 가 비어있는지 여부를 통해 역할을 할당 받을 수 있는지 여부를 판별할 수 있다. 만약 공간이 이미 할당되어 있다면 고유한 역할이거나 부서 이동 등에 따른 역할이 남아 있음을 알 수 있다.

또한 Role Spaces 의 특성을 이용하면 Hierarchical RBAC 을 적용할 수 있다[3]. 부서의 경우 연결된 하위 사용자들의 Role Spaces 가 공통으로 비어있는 공간에 역할을 일괄적으로 할당하면 일반 계층 RBAC 의 조건을 만족하게 된다. 제한 계층 RBAC 또한 Role Spaces 가 공통으로 비어있는 공간에 역할을 할당하려는 사용자들의 역할을 일괄 할당하면 된다. 이렇게 적용하면 <그림 5>와 같이 일반 및 제한 계층 RBAC 상속이 이루어진다.



<그림 5> Role Spaces 를 이용한 상속

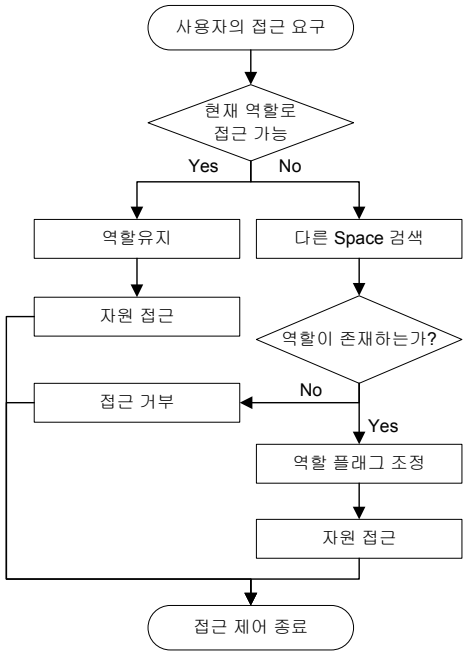
2-2 의무분리(제약)

제약은 NIST 가 지정한 정적 의무분리, MC0 와 동적 의무분리, MC1 을 만족할 수 있어야 한다[2, 3]. 정적 의무분리는 역할을 할당하려고 할 때, 적용하려는

역할 공간 이외의 공간에 이미 할당된 역할과 상충되는 역할의 존재 여부를 통해 의무분리 기능을 수행할 수 있다. 동적 의무 분리는 상충되는 역할이 존재하더라도 실행 시간에 상충되는 역할이 동시에 활성화되는 것을 막는 것으로, Role Space 의 기본 특성인 역할 공간의 활성/비활성 플래그 정보를 통해 실행 시간에 발생할 수 있는 역할의 충돌을 막을 수 있다. 이러한 검증은 정적 의무 분리에서 대부분 수행되지만, 업무의 특성에 의해 충돌하는 역할을 가지고 있더라도 동시에 역할이 활성화되는 것을 막을 수 있다. 또한 동적 의무 분리를 위한 연산 부하를 줄일 수 있다.

2-3 자원의 접근

자원의 접근을 위해서는 해당 자원에 대한 접근 허가가 사용자에게 할당되어 있어야만 가능하다. Role Spaces 는 사용자에게 허용된 역할을 정적으로 포함하고 있으므로, 현재 Role Spaces 를 검색하는 것으로 자원 접근이 가능한 역할의 보유 여부를 쉽게 알 수 있다. <그림 6>은 자원에 접근하고자 하는 사용자가 해당 자원에 대한 접근을 시도할 때, RBAC 이 허가를 허용/거부하는 순서이다.



<그림 6> Role Space 를 이용한 자원 접근

4. 결론 및 연구 방향

본 논문에서는 응용 프로그램 실행 시간에 적용할 수 있는 역할 기반 접근제어를 위한 저장소와 저장소가 NIST 의 표준 참조 모델을 지원하는 방법에 대해 살펴보았다. 제안된 Role Spaces 는 상속과 의무 분리에 대해 적절하게 접근의 허용, 불가 판단을 내리면서 연산 부하를 줄일 수 있고, 한번 구성된 역할의 활용은 비용이 적다는 장점이 있다. 그러나, 정적으로 역할이 할당되어 있어야 하는 단점과 역할이 정적으로 할당되기 때문에 초기 역할의 구성과 할당에 높은 비용이 발생할 수 있다.

앞으로의 연구는 역할이 5 개 이상인 조직에서의 역할 정보의 할당이 필요한 경우의 해결방안이 필요하며, 하나의 Role Space 를 사용하여 구현한 모델과 Role Spaces 를 이용한 모델을 구현하여 각 모델의 성능 비교 및 평가를 통해 좀 더 효과적인 역할 할당에 대한 연구가 필요하다. 그리고, <그림 6>의 다른 Role Spaces 검색 시에 할당된 역할이 해당 자원에 접근할 수 있는 허용된 역할인지를 알아낼 수 있는 효율적인 검색 방법이 필요하다.

마지막으로 Role Spaces 모델은 응용 프로그램 환경을 염두에 두고 작성한 모델이므로, 운영체제의 접근 제어에 그대로 적용하는 것은 문제가 있다. 따라서 설계한 역할 저장소 모델을 운영체제에 적합한 형태로 변경하여 운영체제의 접근 제어로 사용 가능성 여부에 대해 연구하고, 성능 개선을 통해 운영체제에서 사용할 수 있는 접근 제어로 발전시키는 것이 중요한 과제로 남아있다.

참고문헌

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

[2] 김학범, 김동규: RBAC 표준 참조 모델 연구동향. 情報保護學會誌. 2000.

[3] Ravi Sandhu, David Ferraiolo, Richard Kuhn: The NIST Model for Role-Based Access Control; Towards A Unified Standard. 2000.

[4] Wilfredo Alvarez: Presentation on RBAC standard[8].

[5] 유두규, 문봉근, 전문석: PMI 기반의 RBAC 을 이용한 NEIS 의 DB 보안 구현. 情報保護學會論文誌. 2004.

[6] Ravi S. sandhu, Edward J.Coyne, Hal L. Feinstein and Charles E. Youman, "Role-Based Access Control Models," IEEE computer, 1996.

[7] 이철원, 이병각, 김기현, 박정호, 이홍섭, 최용락: 역할 속성을 이용한 역할기반 접근통제 매커니즘. 情報保護學會論文誌. 1998.

[8] http://csrc.nist.gov/rbac/