정적 임무분리를 만족하는 사용자-역할 할당 방안

윤희정° 전준철 김용석 전진우 유기영
경북대학교 컴퓨터공학과
{dude93°, jcjeon33, shadowguys, jwjeon}@infosec.knu.ac.kr yook@knu.ac.kr

Methods of User-Role Assignment for Static Separation of Duty

Hee-Jung Yoon° Jun-Cheol Jeon Yong-Seok Kim Jin-woo Jeon Kee-Young Yoo
Dept. of Computer Engineering, Kyungpook National University

요 약

오래전부터 많은 컴퓨팅 시스템에서 기본적으로 제공 되어지는 원리인 임무분리는 중대한 업무를 둘이상의 사용자에게 나누어줌으로써 단독 사용자가 시스템을 손상시키는 것을 막도록 하는 것이 목적이다. 역할기반 접근통제에서 임무분리 원리를 제공하는 종류로는 정적 임무분리, 동적 임무분리, 기능적 임무분리, 객체기반 임무분리 등이 있다. 여기서 우리는 정적 임무분리를 만족시키는 모델로서 상호 배제 역할 쌍을 이용한 모델과 역할유형을 이용한 모델, 그리고 상호 무관 역할 쌍을 이용한 모델을 제안한다.

1. 서 론

최근 몇 년 동안 컴퓨터의 급성장으로 인해 정보 자원을 유지 관리하는 중·대형 서버급 컴퓨터의 사용이 증가하고 있다. 하지만 불법적인 사람들로부터의 사용, 정보의 노출, 수정, 파괴와 같은 비합법적인 행위로부터 시스템을 보호하기 위해 접근제어가 필요하게 되었다. 여러 연구자들의 노력으로 역할기반 접근제어가 2001년 NIST 표준으로 공표되어 널리 사용되고 꾸준히 향상되고 있다[1]. 역할기반 접근제어에는 역할계층, 임무분리, 권한상속 등 여러 가지 구성요소가 있다. 이중에서 임무분리는 오래전부터 많은 컴퓨팅 시스템에서 제공 되어지는 원리로서 "tow-man rule"로 잘 알려져 있다. 중대한 업무를 둘이상의 사용자에게 나누어줌으로써 단독 사용자가 시스템을 손상시키는 것을 막도록 하는 것이다.

역할기반 접근통제에서 다양한 종류의 임무분리 모델이 제안되었다. 임무분리 원칙을 만족하기 위해서 역할들 사이에는 상호 배제관계가 성립될 수 있는데, 이러한 관계가 성립하는 역할들을 분류하는 강도에 따라 Simon과 Zurko은 임무분리의 방법을 크게 정적 임무분리와 동적 임무분리로 두 개의 범주로 나누었다[2]. 정적 임무분리는 사용자가 역할에 대한 권한을 부여받을 때 역할제약이 적용되는 것이다. 예를 들어, 역할 A, B가 상호 배제관계일 때, 한 사용자가 역할 A에 대한 권한을 부여받았다면, 같은 사용자가 역할 B에 대한 권한은 부여받지 못하는 정책이 필요하다. 반면, 동적 임무분리에서는 한 사용자가 역할 A와 역할 B 둘다의 권한을 부여받을 수 있으나, 단독 작업시간에서는 두 역할을 동시에 유지할 수 없다[3].

이외에도 객체기반 임무분리, 기능적 임무분리, 히스토리 기반 임무분리 등이 있는데 이들은 약한 배제관계로 동적 임무분리의 범주에 포함된다[4-6]. 반면 정적 임무분리는 개념이 명백하고 간단하기에 변형된 모델이 없다. 따라서 본 논문에서는 이러한 정적 임무분리를 실제 시스템에 적용하기 위한 세 가지 모델을 제안한다. 먼저 상호 배제관계의 역할 쌍을 이용해 사용자-역할 할당 시에 상호 배제관계의 역할이 할당되어 있는지를 검사하는 모델을 제시하고, 역할 쌍을 이용한 정적 임무분리 모델을 개선한 역할유형을 이용한 정적 임무분리 모델을 제안한다. 마지막으로 시스템 상에 상호 배제관계의 역할 쌍이 매우 많은 경우에 적용할 수 있는 상호 무관 역할 쌍을 이용한 정적 임무분리 모델을 제안한다.

2. 정적 임무분리 원칙을 만족하는 사용자-역할 할당 방안

본 절에서는 정적 임무분리 원칙을 만족시키기 위한 정책 중 사용자-역할 할당에 관한 세 가지 구현방안들과 그에 따른 제약사항들을 제안한다.

2.1 상호 배제 역할 쌍을 이용한 정적 임무분리

임무분리 원칙을 만족하기 위해 역할들 사이에 존재하는 상호 배제관계를 역할 쌍을 이용해 표현할 수 있다. 다음은 상호 배제관계를 가진 네 개의 역할들(R1,…,R4)을 나타낸 예이다.

상호 배제 역할 쌍 = { (R1, R2), (R1, R3), (R2, R3), (R3, R4) }

상호 배제관계의 역할 쌍들이 존재하고, 역할을 사용자에게 할당할 때, 그 역할과 상호 배제관계를 가지는 역할을 역할 쌍의 집합에서 찾는다. 해당 역할이 역할 쌍 집합에 존재하면, 상호 배제관계의 역할이 사용자에게 이미 할당되어 있는가를 확인하여 할당되지 않았다면 그 역할을 사용자에게 할당한다. 그림 1은 사용자-역할 할당에 역할 쌍을 이용하는 정적 임무분리 모델이다.

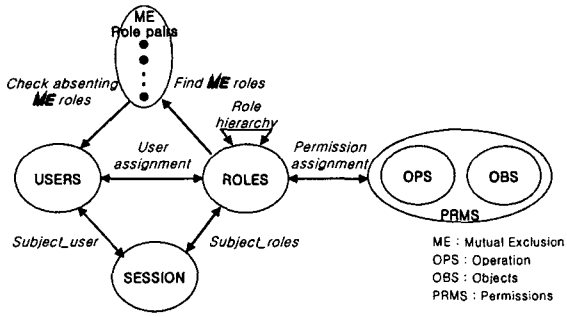

그림 1 상호 배제 역할 쌍을 이용한 정적 임무분리 모델

역할 쌍을 이용한 정적 임무분리 모델에서는 사용자-역할 할당 시에 할당하고자 하는 역할을 역할 쌍만큼 비교해야 한다. 전체 역할의 수를 $n(R)$ 이라 할 때, 최악의 경우 즉, 모든 역할들이 상호 배제관계를 가진다면 $n(R)\times(n(R)-1)/2$ 만큼의 역할 쌍을 가지게 되고, 따라서 상호 배제관계 역할의 존재검사에 따른 시간복잡도가 증가하게 되는 문제점을 가진다. 이 모델의 문제점을 개선하기위해 다음절에서 역할유형을 이용한 모델을 제안한다.

2.2 역할유형을 이용한 정적 임무분리

역할유형을 이용한 사용자-역할 할당방법은 역할 쌍을 이용한 방법을 더 개선시킨 것으로서 역할유형이라는 개념을 사용하여 역할을 분류한다. 역할유형이란 상호 배제관계의 역할을 구분할 수 있는 값이고, 각각의 역할은 하나의 역할유형을 가진다. 상호 배제관계에 따라 나누어진 역할집합들은 서로 다른 역할 유형을 가지게 되고, 하나의 역할집합내의 역할들은 동일한 역할유형을 가진다. 만약 모든 역할들이 상호 배제적인 관계를 가진다면 역할유형은 역할의 수만큼 생성된다. 역할 쌍의 표현만으로는 몇 개의 역할유형이 필요한지, 또 어떤 역할들이 같은 역할유형으로 설정될지 결정하기가 어렵다.

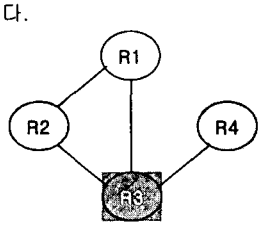

그림 2 정점채색 이론을 적용한 상호 배제관계의 그래프

따라서 여기에 그래프 이론 중 정점채색 이론을 이용하여 역할들 사이의 상호 배제관계를 고려한 역할유형의 수를 결정할 수 있다. 그림 2는 2.1절에서의 상호 배제관계의 역할 쌍들을 그래프로 나타낸 것이다. 각 역할들은 정점이 되고, 상호 배제관계의 역할들은 서로 간선으로 연결된다.

정점채색 이론은 인접한 정점들이 서로 다른 색을 가지도록 그래프의 각 정점에 한 가지 색을 할당하는 것으로 채색수 $\chi(G)$는 그래프를 채색하는데 필요한 최소의 색의 가지수를 의미한다[7]. 채색 수 $\chi(G)$를 구함으로서 역할들 사이의 상호 배제관계를 고려한 역할유형의 수가 결정되어 진다. 그림 2와 같이 4개의 정점은 세 개의 채색 수를 가지므로 역할유형의 수는 3으로 결정되고, 역할들은 세 개의 역할집합({R1}, {R2, R4}, {R3})으로 분류된다.

각 역할의 역할유형이 결정되면 사용자에게 역할을 할당한다. 이때 사용자는 하나 이상의 역할을 할당 받을 수 있으나 한 가지 유형의 역할에 대해서만 권한을 부여받을 수 있다. 역할유형은 사용자-역할 할당에 있어서 한 사용자에게 같은 유형의 역할들만 할당이 가능하다는 제약조건으로 이용되며, 역할계층에 있어서도 같은 역할유형을 가진 역할그룹 내에서만 상속이 가능하다는 제약조건으로 이용된다. 같은 역할유형을 가진 역할집합 내에서의 역할상속은 역할계층상에서 역할상속으로 인한 임무분리 원칙의 파괴가능성을 제거할 수 있다. 그림 3은 역할유형을 이용한 정적 임무분리 모델을 나타낸다.

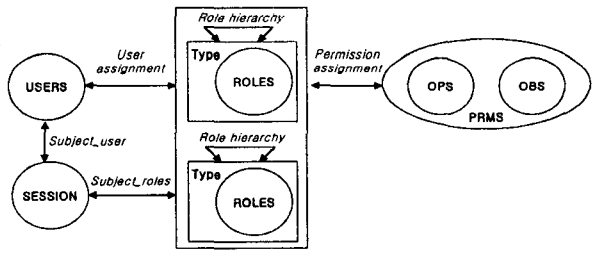

그림 3 역할유형을 이용한 정적 임무분리 모델

역할 쌍을 이용한 정적 임무분리 모델에서 최악의 경우, 사용자에게 하나의 역할을 할당할 때 $n(R)\times(n(R)-1)/2$ 만큼의 탐색이 필요하다. 그러나 역할유형을 이용할 경우 처음 역할유형을 결정할 때의 정점색채 이론의 적용시간을 제외하면 역할유형이 동일한지만 비교하면 되므로, 제약조건의 검사는 한번만 이루어진다.

2.3 상호 무관 역할 쌍을 이용한 정적 임무분리

어떤 시스템에서는 상호 배제관계의 역할 쌍들이 매우 많을 수 있다. 상호 배제관계의 역할 쌍의 수가 상호 무관한 역할 쌍의 수보다 많다면, 그림 1의 상호 배제관계의 역할 쌍을 이용하는 것 보다 상호 무관한 역할 쌍을 이용해 정적 임무분리를 만족시키는 것이 더 효율적이다. 이 모델을 적용시키는 경우는 다음과 같이 제한한다.

$n(MERP)$: the number of Mutual Exclusion Role pair

$$n(R)\times(n(R)-1)/4 \;<\; n(MERP) \;<\; {}_{n(R)}C_2$$

상호 무관한 역할들의 쌍이 존재하고, 사용자-역할 할당을 할 때, 우선 사용자에게 처음 역할을 할당할 때는 제약조건 없이 한 개의 역할을 할당할 수 있다. 한 사용자에게 두 개 이상의 역할을 할당할 때부터 제약조건이 적용되는데, 제약조건은 할당하고자 하는 역할이 상호 무관 역할 쌍에 존재한다면 그 역할과 쌍을 이루는 역할이 사용자에게 이미 할당되어 있는가를 검사하는 것이다. 그림 4는 상호 무관 역할 쌍을 이용한 정적 임무분리 모델을 나타낸다.

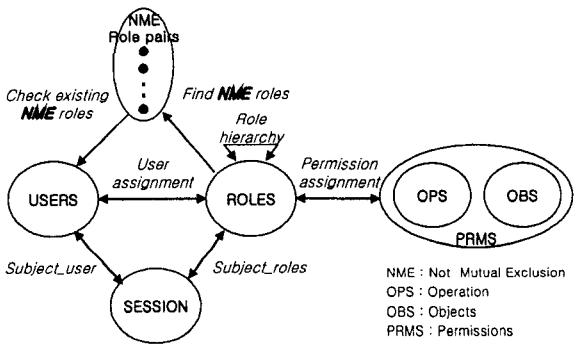

그림 4 상호 무관 역할 쌍을 이용한 정적 임무분리 모델

3. 제안된 모델의 분석

본 절에서는 제안한 세 가지 모델을 전체 역할의 수와 상호배제 관계의 역할 쌍의 수에 따른 차이점을 비교한다.

표 1 다양한 조건에서의 각 정적 임무분리 모델의 비교

조건 / 모델	역할의 수 = n(R)					
	50		100		200	
	n(MERP)		n(MERP)		n(MERP)	
	최소	최대	최소	최대	최소	최대
그림 1	1	1225	1	4950	1	19900
그림 4	1224	0	4949	0	19899	0

전체 역할의 수가 50, 100, 200개인 세 가지 경우에 대해 각각 상호 배제관계의 역할 쌍의 수가 최소일 때와 최대일 때를 구분하여 각 모델들을 적용하였다. 각 경우마다 상호 배제관계의 역할 쌍은 적어도 하나 이상 존재한다고 가정한다. 표 1 안의 수치는 한 역할이 사용자에게 할당될 때, 정적 임무분리를 만족하기위해 고려되어야 할 제약조건의 적용을 위한 탐색횟수를 나타낸다. 그림 1(상호 배제 역할 쌍을 이용한 정적 임무분리 모델)의 경우, 상호 배제 역할 쌍의 수와 동일한 값을 가진다. 그림 4(상호 무관 역할 쌍을 이용한 정적 임무분리 모델)의 경우는 첫 번째 모델과 반비례한 값을 가지게 된다. 이 모델이 적용될 수 있는 바람직한 경우를 예를 들면 전체 역할의 수가 50일 때 상호 배제 역할 쌍의 수가 613 이상 1225 이하인 경우이다.

그림 3(역할유형을 이용한 정적 임무분리 모델)의 경우에는 각 역할의 역할유형을 결정하는 시간을 고려해야 한다. 역할유형 결정은 처음에 한번만 수행되고, 수행에 필요한 시간은 상호배제역할 쌍의 수와 비례한다. 일단 역할유형이 결정되어지면 역할유형이 동일한지만 비교하면 되므로, 제약조건의 검사는 전체 역할의 수나 상호 배제관계의 역할 쌍의 수와 상관없이 한번만 검사하면 된다.

4. 결론

역할기반 접근통제에서 임무분리는 반드시 만족되어야 하는 원칙 중 하나로, 이것을 만족시키기 위한 다양한 모델이 연구되어왔다. 본 논문에서는 정적 임무분리를 만족시키기 위한 실제적인 모델로 상호 배제 역할 쌍을 이용한 모델과 역할유형을 이용한 모델, 그리고 상호 무관 역할 쌍을 이용한 모델을 제안하고 분석하였다. 시스템 결정적인 조건들을 앞서 제시한 경우들과 비교하여 시스템에 알맞은 모델을 선택하여 적용시키면 효과적으로 임무분리 원칙을 만족시키고 성능을 향상 시킬 수 있을 것으로 기대한다.

참고문헌

[1] National Institute of Standards and Technology, *Proposed Standard for Role-Based Access Control*, http://csrc.nist.org/rbac/rbacSTD-ACM.pdf , 2001.

[2] Simon, R. T., and M. E. Zurko, "Separation of Duty in Role Based Environments," *Proc. Computer Security Foundations Workshop X*, pp. 183-194, June 1997.

[3] David F. Ferraiolo, D. Richard Kuhu, Ramaswamy Chandramouli, *Role-Based Access Control*, ARTECH HOUSE, INC. 2003.

[4] Nash, M. J., Poland, K. R. "Some Conundrums Concerning Separation of Duty," *Proc. 1990 IEEE Symposium on Security and Privacy*, pp. 201-207, May 1990.

[5] Ferraiolo, D., Cugini, J., Kuhn, D. R. "Role-Based Access Control (RBAC): Features and Motivations," *Proc. 1995 Computer Security Applications Conference*, pp. 241-248, Dec. 1995.

[6] Sandhu, R. "Transaction Control Expressions for Separation of Duties," *Proc. 4th Aerospace Computer Security Conference*, pp. 282-286, Dec. 1998.

[7] http://mathworld.wolfram.com/VertexColoring.html