역할-기반 접근 제어 모델에서 역할 위임을 위한 프로토콜 설계

나 상엽*, 전 서현**
**동국대학교 컴퓨터 공학과
*남서울대학교 컴퓨터학과, nsy@www.nsu.ac.kr

Role Delegation Protocol for Role-Based Access Control

Sangyeob Na*, Seohyun Cheon**
**Department of Computer Engineering, Dongguk University
*Department of Computer, Namseoul University

요 약

분산 컴퓨팅 환경에서 기업이나 조직내의 사용자들은 다른 사용자와 자원을 공유하며 상호작용을 통하여 보다 효율적으로 작업을 수행하게 된다. 이 경우 공유하는 자원이나 정보의 불법적인 사용을 막고 데이터의 무결성을 유지하기 위하여 사용자의 인증과정이 필요하며, 또한 사용자의 작업에 대한 접근 제어(Access Control)의 필요성이 더욱 중요시되고 있다.

현재 널리 알려진 임의적 접근 제어(DAC)는 객체의 소유자에게 모든 임의의 권한이 주어지고 강제적 접근 제어(MAC)의 경우에는 주체와 객체단위의 정책 적용이 어려운 단점이 있다. 최근에는 역할-기반 접근 제어를 이용하여 조직의 보안 정책을 보다 효율적이고 일관성 있게 관리하고자 하는 시도가 있다. 하지만 역할-기반 접근 제어의 경우 각 역할의 계층에 의하여 권한의 상속이 결정되는 문제가 발생한다. 따라서 본 논문에서는 역할-기반 접근 제어에서 역할이 가지는 역할의 위임을 위한 위임 서버와 역할 위임 프로토콜을 제시한다.

1. 개 요

분산 컴퓨팅환경이 보편화됨에 따라 기업이나 조직내의 사용자들은 서로의 자원을 공유하며 상호 작용을 통하여 보다 효율적으로 작업을 수행한다. 여러 사용자들이 상호 작용하는 분산 컴퓨팅 환경에서 공유하는 자원이나 정보가 증가함에 따라 허가되지 않은 정보의 접근이 발생할 수 있고 불법적인 사용으로 인한 정보의 누출이 발생한다. 따라서 분산 컴퓨팅 환경의 정보를 보호하기 위하여 사용자의 인증이나 사용자의 작업에 대한 접근 제어 정책을 통한 정보 보안의 필요성이 증가하고 있으며 이러한 접근 제어 정책은 시스템 사용의 편리성을 위하여 사용자나 응용프로그램에 투명하게 제공되어야 한다[5].

접근 제어 정책은 정보의 소유자에 의하여 접근 통제 관계가 정의되는 임의적 접근 제어(Discretionary Access Control : DAC), 정보의 내용(보안등급)과 사용자나 그가 속한 그룹에 의하여 접근을 제어하는 강제적 접근 제어(Mandatory Access Control : MAC), 그리고 시스템 내에 필요한 역할(Role)과 그 역할이 수행할 수 있는 권한(연산)을 정의하고 각 사용자에게 역할을 할당하는 역할-기반 접근 제어(Role-Based Access Control : RBAC)기법으로 나눌 수 있다[2][7][8]. 강제적 접근 제어의 경우 정보(객체)의 보안 등급에 의한 접근 제어를 수행하므로 각각의 사용자와 객체단위로 접근 제어의 설정이 불가능하고 임의적 접근 제어의 경우 객체의 소유자에 의한 접근 정책의 임의변경과 다른 주체에게 자신의 허가된 권한의 임의위임이 가능하여 정보의 효율적 제어가 어렵다. 이에 비하여 역할-기반 접근 제어의 경우 미리 정의된 역할, 역할이 수행할 수 있는 접근 권한(Permission)을 명시하고 사용자에게 역할을 부여하므로 사용자는 자신에게 할당된 역할에 의하여 객체를 접근할 수 있다. 따라서 조직은 조직의 특성에 적합한 접근제어 정책을 일관성 있게 유지할 수 있을 뿐 아니라 주체와 자원의 접근 권한 관계를 독립적으로 유지하므로 접근 권한이 변경될 때 새로운 권한을 사용자가 아닌 역할에만 적용하면 되고 복잡한 보안 정책도 추상화하여 효율적으로 관리할 수 있다.

역할-기반 접근 제어에서 역할은 조직 내에서 객체에의 접근이 허가된 권한과 책임들의 집합으로 볼 수 있다[1]. 역할-기반 접근 제어에서 각각의 역할은 조직의 접근 정책에 따라 다른 역할과의 상관관계를 가지고 계층구조로 표현되며 상위개념의 역할은 하위개념의 역할의 권한을 상속(Inheritance)받는다[7][8]. 이 경우 하위개념의 역할이나 계층구조로 포함되지 않은 역할은 상위개념의 역할이 가지는 권한을 수행할 수 없는 문제가 있다. 본 논문에서는 역할의 위임을 통하여 하위개념의 역할이 일시적으로 상위 개념의 역할이 가지는 권한을 수행할 수 있는 방법을 제시하며 위임의 방법을 자신이 다른 역할의 권한을 위임받는 능동적 위임과 다른 사용자에게 역할의 위임을 수행하는 수동적 위임으로 구분한다. 또한 위임의 결정을 내리는 위임서버와 위임을 위한 간략화된 위임 프로토콜을 제시한다. 최근의 역할-기반 접근 제어는 의료기관의 정형화된 모델을 제시하기 위한 연구가 활발하므로[9] 이 논문에서 사용하는 예도 의료기관의 모델을 중심으로 기술하도록 한다.

본 논문의 구성은 2장에서 역할-기반 접근 제어의 개념과 모델을 소개하고, 3장에서는 역할이 가지는 권한의 위임, 위임 정책, 위임 서

버, 그리고 위임정책 프로토콜을 제시하고 4장에서 결론 및 향후 연구를 기술한다.

2. 역할-기반 접근 제어(RBAC)

접근 제어 정책(Access Control Policy)은 식별 또는 인증된 사용자가 허가된 범위 안에서 시스템 내부 정보로의 접근을 허용하는 방법을 기술한다. 사용자는 자신이 가진 접근 허가권에 의하여 정보에의 접근이 허가 또는 거부된다. 이때 사용자는 주체(Subject)가 되고 사용자가 접근을 원하는 정보는 객체(Object)가 된다. 객체는 사용자가 가지는 권한에 따라 접근이 허용되므로 동일한 객체라도 역할에 따라 수행할 수 있는 연산은 달라진다.

기존의 임의적 접근 제어의 경우 객체는 사용자가 소유하게 되고 객체의 소유자는 객체의 모든 접근 권한을 가지고 다른 사용자에게 임의로 접근 정책을 부여하고 강제적 접근 제어는 조직내의 관리자가 객체의 중요도에 따라 접근 권한을 사용자에게 할당하여 준다. 이에 비해 역할-기반 접근 제어에서는 시스템 관리자가 역할에 접근 권한을 부여하고 사용자는 자신의 책임과 권한에 따라 역할을 부여받으므로 복잡한 조직의 형태를 보다 효율적으로 표현할 수 있다[8].

2.1 역할-기반 접근 제어 개념

역할-기반 접근 제어에서 역할은 조직 내에서의 권한과 의무의 집합으로 시스템 관리자에 의하여 사용자에게 할당된다. 이는 그림 1로 표현되며[6] 역할에는 필요에 따라 권한이 부여되거나 삭제될 수 있다. 역할-기반 접근 제어 모델의 기본 구성요소로는 사용자(User), 역할(Role), 행위(Action), 그리고 권한(Permission)등이 있다.

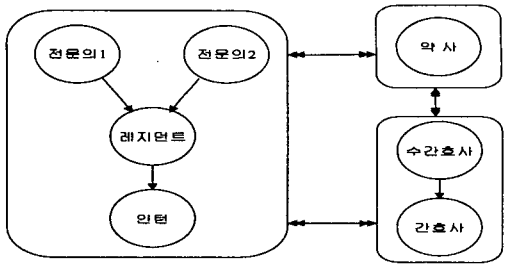
[그림 1] 역할-기반 접근 제어의 기본 모델

사용자는 시스템내의 응용프로그램이나 사람을 나타내고, 역할은 조직 내에서 권한과 의무를 가지는 직위를 표현한다. 권한은 역할이 하나 이상의 객체에 접근할 수 있는 방법을 나타내며 의무, 허가 등으로 세분화되어지며[1] 역할도 특정 기능을 가지는 객체로 표현된다. 행위는 역할이 특정 객체에 수행할 수 있는 연산의 집합으로 표현된다. 역할-기반 접근 제어에서 사용자는 하나 이상의 역할에 할당되며 사용자는 자신에게 할당되어진 역할이 가지는 권한에 따라 객체에 연산을 수행한다. 역할-기반 접근 제어에는 사용자-역할(UA) 관계, 역할-권한(RP) 관계, 역할-역할(RR)의 관계가 존재하며 본 논문에서는 역할 위임의 설명을 위해 역할-역할 관계와 역할-권한관계를 주로 설명한다.

2.2 역할-역할 관계(Role-Role relationship)

역할-기반 접근 제어 모델에서 역할은 조직 내에서 역할의 책임과 권한 등에 따라 계층구조(Hierarchical Structure)로 표현되며 유사한 권한을 가지는 역할들은 그룹으로 관리된다[5]. 그룹은 역할의 관리를 위하여 조직내의 관리자나 조직의 구성에 의하여 분리된다. 역할은 역할 그룹 내 역할 계층구조에서 역할의 위치에 따라 상위 역할과 하위 역할로 구분되며 상위 역할은 하위 역할의 권한을 가지는 상속관계와 다중 상속 관계도 성립한다. 또한 서로 이웃하는 역할 그룹간에 연산관계도 존재한다[4].

그림 2의 경우에서처럼 병원 조직의 의사 그룹 내의 인턴이 가지는 환자에 대한 권한은 상위 역할인 레지던트와 전문의에게 상속되며 의사 그룹과 간호사 그룹 그리고 약사는 상호간에 연산관계가 존재한다.



[그림 2] 역할 그룹과 상속관계, 그룹간의 연산 관계

2.3 역할-권한 관계(Role-Permission relationship)

권한은 의무와 허가로 나뉘는데 의무는 해당 역할이 반드시 수행하거나 하지 말아야 하는 연산의 집합이고 허가는 역할에게 허용되거나 허용되지 않은 연산의 집합이다. 의무와 허가의 표현은 [4]에서 정의한 방법을 단순화하여 사용하도록 한다.

{식별자, 모드, 역할, {행위}, 대상, 조건, 예외}

[모드] o : 의무(Obligation), a : 허가(Authorization)
+ : positive, - : negative

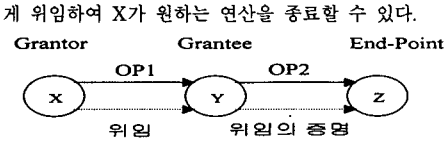
예를 들어 "간호사는 매일아침 8시 환자의 상태를 체크하여야 한다"라는 의무는 **{np1, o+, 간호사, {체크}, 환자, 매일08:00, -}**와 같이 표현될 수 있고, "전문의는 인턴이 기록한 환자차트를 볼 수 있다"의 허가는 **{dp1, a+, 전문의, {read}, 차트by인턴, -, -}**로 표현된다. 본 논문에서 사용하게되는 역할-권한 관계의 예는 표1과 같다.

역할 그룹	권 한
의 사	{dp1, a+, 전문의, {read, fix}, 차트by인턴, -,-}
	{dp2, a-, 인턴, {조제}, 약, -, 응급}
간 호 사	{np1, o+, 수간호사, {배정}, 간호사-환자, 매일09:00,-}
	{np2, a+, 간호사, {주사by차트}, 환자, 매일12:00, -}
약 사	{drp1, a+, 약사, {조제by차트}, 환자 , -, -}
	{drp2, o+, 약사, {수량정리}, 사용한약, 매일퇴근시, -}

[표 1] 간단한 역할-권한 관계 모델

3. 위임 정책

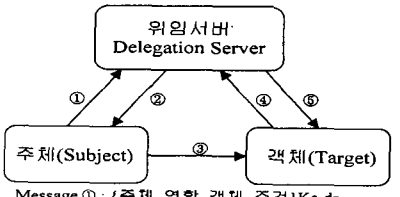
역할-기반 접근 제어 정책을 사용하는 시스템의 경우 역할 또는 역할에 주어진 권한을 다른 역할이나 다른 사용자에게 위임 또는 전가를 필요로 하는 경우가 발생한다[5][10]. 즉 사용자가 할당받은 자신의 역할을 다른 사용자에게 위임하여 위임을 받은 사용자가 해당 역할의 권한을 수행하게 되거나 그림 3의 경우처럼 X가 Y에게 수행한 OP1 연산이 Y가 Z에게 행하는 OP2 연산의 계기가 되는 경우가 일어날 수 있다. 만약, X가 Z에 대하여 OP2 연산의 권한을 가지고 있고, Y는 Z에게 OP2 연산을 수행할 수 있는 권한을 가지고 있지 않더라도 Y의 OP2 연산은 궁극적으로 X의 OP1연산에 의한 것이므로 Y에게 일시적으로 OP2를 수행할 수 있도록 X가 가지는 OP2의 권한을 Y에게 위임하여 X가 원하는 연산을 종료할 수 있다.



[그림 3] 위임 정책의 기본 모델

3.1 역할 위임 서버와 프로토콜

역할 위임 서버는 역할-역할 관계, 역할-권한 관계, 역할 계층, 그리고 예외 조건 등을 종합하여 역할 위임의 가능 여부를 최종적으로 결정해주는 기능을 수행한다. 역할 위임이 필요한 경우 위임을 원하는 주체(Subject)는 위임을 원하는 역할과 대상을 명시하여 위임 서버에게 위임 증명의 발급을 요구한다. 이 정보는 자신과 위임 서버 사이에 공유하는 비밀키로 암호화하여 전달한다, 위임 서버는 주체가 전달한 정보를 자신이 가지는 비밀키로 복호화 후에 자신이 가지는 하여 위임 가능 정보와 비교하여 위임 가능 여부를 판단하고 위임서(Certificate)를 다시 암호화하여 주체에게 전달한다. 주체는 서버로부터 받은 위임서를 자신과 객체 사이의 공유키로 암호화하여 객체에게 전달한다. 위임서를 전달받은 객체는 이를 위임 서버에게 보내 검증을 요구하고 서버는 위임서가 올바른 것이라는 것을 증명하여 준다. 위와 같은 과정을 거치면 객체는 주체에 의하여 해당 역할을 위임받을 수 있으며 역할-권한 관계에 명시되어진 권한을 수행한다. 이때 위임서에는 위임이 유효한 시간을 명시하여 주어진 시간만큼 객체가 일시적으로 역할의 권한을 수행하게 한다. 프로토콜은 그림4로 표현되고, 위임서버와 주체, 객체 사이에는 공개키 와 비밀키 알고리즘 [11]이 사용된다고 가정한다.



Message ① : {주체, 역할, 객체, 조건}Ks-ds
(Delegate Request)
Message ② : {Certificate}Ks-ds {Certificate }Kds
(Delegation Request Result : Certificate)
Message ③ : {Certificate}Kds
(Certificate)
Message ④ : {객체, {Certificate}Kds}Kt-ds
(Certificate Authentication)
Message ⑤ : {Certificate}Kt-ds
(Authentication Result)

Certificate : <객체, 역할, 기간>
Ki-j : i 와 j 가 공유하는 비밀키
Kk : k의 비밀키

[그림 4] 역할 위임 프로토콜

3.2 능동적 역할 위임 과 수동적 역할 위임

역할 위임은 자기자신이 해당 역할의 위임을 요청하는 능동적 위임과 기본 모델처럼 다른 역할의 연산으로 인하여 역할을 위임받는 수동적 위임으로 나눌 수 있다.

능동적 위임의 경우 역할-권한 관계에 a- 모드를 가지면서 예외 필드가 존재하는 경우인데 이 경우 예외 조건이 만족하면 자신이 위임 서버에게 해당 역할의 위임을 요청한다. 위임 서버는 위임을 위한 모든 정보를 수집하여 해당 역할의 위임 여부를 결정한다.

2.3의 예에서 의사 그룹과 간호사 그룹은 약을 조제하는 권한이 주어지지 않는다. 의사 그룹 중 인턴의 경우 역할-권한 관계 내에 **{dp2, a-, 인턴, {조제}, 약, -, 응급}** 권한을 가지고 있으며, 예외 조건(응급)이 명시되어 있으므로 예외 조건이 만족하는 경우 모드 a- 는 a+로 변환되어 약사의 권한을 위임받을 수 있다. 또한 의사 그룹에서 인턴의 상위 개념인 레지던트, 전문의의 경우 하위 역할의 권한을 상속받으므로 예외조건이 만족하면 약사의 권한을 위임받을 수 있다.

수동적 위임의 경우 역할-권한 관계에 위임 대상 역할의 권한이 명시되지 않은 경우인데 이러한 경우에는 역할-역할 관계에 의하여 동일 그룹 내에서 상위 역할의 권한을 하위 역할이 위임받거나 그룹간의 연산관계에 의하여 다른 그룹의 역할에 권한을 위임하는 경우이다. 이때는 그룹간 연산 관계에 존재하는 제약 조건이나 역할의 자격(qualification) 존재 여부 등에 의하여 위임 여부가 결정된다.

간호사의 경우 의사 역할을 가지는 사용자가 간호사에 약사의 역할이 필요하다고 판단되면 위임 서버에게 {의사, 약사, 간호사, 응급}를 이용하여 위임 요청을 하고 위임 서버는 위임 여부를 판단하여 위임서를 의사에게 전달한다. 의사는 위임 서버로부터 받은 위임서를 간호사에게 전달하여 간호사가 제한된 시간동안 약사의 역할을 수행할 수 있도록 한다.

4. 결론 및 향후연구

본 논문에서는 현재 활발한 연구가 진행되고 있는 역할-기반 접근 제어 개념을 살펴보고 이의 간단한 모델을 정의하였다. 역할-기반 접근 제어 모델의 경우 복잡화된 조직의 접근 제어를 정형화하여 표현할 수 있으며 접근 제어 정책의 변화 시에 사용자-권한의 변환이 아닌 역할-권한의 관계를 수정하면 되므로 정책 변환에 보다 융통성 있게 적용할 수 있다. 그러나, 역할의 계층구조에 의한 권한의 상속이 정적이고 조직내의 상호 작용에서 역할이 갖지 않는 권한에 대한 연산을 수행할 필요가 있는 경우 기존의 역할-기반 접근 제에 모델에는 적용하기 어려운 문제가 발생한다.

본 논문에서는 이러한 문제를 해결하기 위하여 일시적으로 필요한 역할을 수행하기 위한 프로토콜과 위임 여부를 결정하는 위임서버를 제시하였으며 이를 해결하였으며 위임정책을 능동적 위임과 수동적 위임으로 분리하여 적용하였다.

향후 본 논문에서 제시한 역할 위임을 보다 세분화하여 역할의 위임뿐 아니라 역할-권한 관계의 제한된 권한의 위임을 위한 프로토콜의 설계와 위임 서버의 기능을 수행하기 위한 정보의 정형화된 표현이 필요하다.

참고 문헌

[1] E. C. Lupu, D. A. Marriott, M. S. Sloman, and N. Yialelis, "A Policy Based Role Framework for Access Control", First ACM/NIST Role Based Access Control Workshop, Dec, 1995

[2] Department of Defence(USA), Department of Defence Trusted Computer System Evaluation Criteria, DoD 5200-78-STD, DoD, 1985

[3] L. Giuri, "Role-Based Access Control in Java", 3rd ACM Role-Based Access Control Workshop, 1998.

[4] E. C. Lupu, M. S. Sloman, "A Policy Based Role Object Model", Proceeding of IEEE EDOC'97, Oct, 1997.

[5] N. Yialelis, M. S. Sloman, "A Security Framework Supporting Domain Based Access Control in Distributed Systems", ISOC Symposium on Network and Distributed System Security(SNDSS96), Feb 1996

[6] David F. Ferraiolo and Richard Kuhn, "Role-based access control," Proceedings of the 15th NIST-NSA National computer security conference, 1992

[7] Ravi S. sandhu, Edward J.Coyne, Hal L. Feinstein and Charles E. Youman, "Role_-Based Access Control Models," IEEE computer, Volume 29, number 2, Feb 1996

[8] David F. Ferraiolo, J. Cugini and Richard Kuhn, "Role-Based Access Control: Features and Motivations," National Institute of standards and technology, 1995

[9] J. Barkley, "RBAC in Health Care", 1995 http://hissa.ncsl.nist.gov/rbav/

[10]C. Goh, A. Baldwin, "Towards a more Complete Model of Role", 3rd ACM Role-Based Access Control Workshop, 1998.

[11]B, Lampson, M, Abadi, and R, Needham, "A Logic of Authentication", ACM Transaction on Computer System, Vol. 8(1), 1990