강제적 접근방식과 역할 그래프를 기반으로 한 역할관리 보안모델

박기홍, 김응모
성균관대학교 전기전자및컴퓨터공학과
E-mail : hongyi@ece.skku.ac.kr

Role Administration Security Model based on MAC and Role Graph

Ki-Hong Park, Ung-Mo Kim
Dept. of Electric, Electronic and Computer Science,
SungKyunKwan University

요 약

다중등급을 갖고 있는 대용량 데이터베이스 환경에서 각 보안등급을 갖고 있는 사용자가 데이터베이스에 접근할 때 확장된 강제적 접근제어(MAC:Mandatory Access Control) 방식과 역할 그래프(Role Graph)를 이용해 하위등급의 사용자가 상위등급의 데이터를 추론하거나 인지하는 데이터 유출을 방지하여 데이터의 무결성(integrity)과 데이터베이스 관리시스템(DBMS:Database Management System) 전체의 보안을 유지하며 각 보안등급의 데이터와 사용자를 효율적으로 관리하고 제어할 수 있는 역할관리 보안모델을 제안한다.

1. 서론

1970년대 초반에 들어서면서 컴퓨터 시스템은 다양한 응용 프로그램을 한 사용자가 아닌 다수의 사용자들에게 제공하는데 초점을 맞추게 되었다. 이 시점에서 보안에 대한 관심이 증대되기 시작했다. 시스템 관리자 입장에서는 시스템 자원을 권한이 있는 사용자에게 정당한 접근정책을 사용해 제공하는데 중점을 두었다. 또한 통신망의 발달에 따라 개인이 접하게 되는 정보의 양은 계속적으로 증가하고 있으며 이를 유지하기 위해 데이터베이스의 규모도 기하급수적으로 증가하고 있다. 데이터베이스 측면에서도 다양한 등급을 갖는 다수의 사용자가 접근을 하게 되었고 데이터베이스 관리자의 측면에서는 보안의 유출의 일어나지 않으면서 다양한 사용자의 요구를 충족시킬 수 있는 서비스를 제공하는데 중점을 두고 있다.

컴퓨터 시스템과 데이터베이스에서 권한이 없는 사용자들이 시스템에 접근하여 정보를 얻거나 데이터베이스 일부를 악의적으로 변경하는 것을 제어하는 것을 접근제어(Access Control)라 한다. 접근제어 방식은 크게 세 가지로 구분할 수 있다. 첫째, 임의적 접근제어(DAC:Discretionary Access Control) 방식. 둘째, 강제적 접근제어(MAC:Mandatory Access Control) 방식. 그리고 마지막으로 역할기반 접근제어(RBAC:Role Based Access Control) 방식이다[1].

본 논문에서는 데이터베이스 환경에서 전통적인 역할 모델에 격자(Lattice) 모델[2]이라고 불리는 MAC 모델을 확장 적용해 다중등급을 갖는 관계형 데이터베이스에서 보안의 누수없이 데이터와 사용자를 효율적으로 관리하고 제어할 수 있는 역할관리 보안모델(Role Administration Security Model)을 제안하는데 중점을 두었다.

본 논문의 구성을 살펴보면 다음과 같다. 2장에서는 관련연구로서 DAC, MAC과 RBAC의 개념을 소개하며, 3장에서는 확장된 MAC과 Role Graph를 이용한 역할관리 보안모델을 제시하며 4장에서는 사례를 이용한 Role Graph를 표현한다. 마지막 5장에서는 결론을 제시한다.

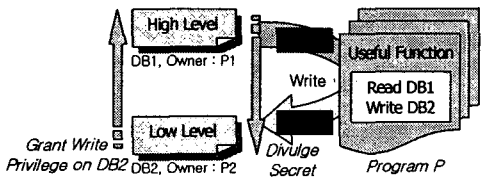
2. 관련연구

데이터베이스 보안의 목적은 서로 상호관계를 맺고 있는 비밀성(secrecy), 무결성(integrity), 그리고 가용성(availability)으로 구분할 수 있다[3]. 첫째, 비밀성 혹은 기밀성(confidentiality)은 정보의 부적절한 누출에 관심을 갖는다. 둘째, 무결성은 정보나 프로세스의 부적절한 수정에 관심을 갖는다. 마지막 가용성은 정보 처리에 있어 부적절한 접근 거부에 관심을 갖는다. 이것은 서비스 거부로 표현할 수 있다.

본 장에서는 다중등급(multilevel)을 갖는 데이터베이스 접근제어 관련연구로서 접근제어 방식인 DAC, MAC, RBAC의 특징을 살펴보겠다.

2.1 DAC(Discretionary Access Control)

DAC은 주체(subject)의 신분을 기반으로 하는 접근 통제로서 일반적인 상업용 관계형 데이터베이스에서 널리 사용되고 있다. 한 사용자가 임의의 객체(object)에 접근할 수 있는 자신의 권한을 다른 사용자에게 허가(grant) 할 수 있다. DAC의 단점은 사용자가 시스템이나 데이터베이스에 접근할 때 인가받은 접근이라 할지라도 제한을 우회할 수 있다는 것이다. 하위등급의 사용자가 상위등급의 사용자가 접근하는 데이터에 접근할 수 있다는 것이다. 즉 DAC은 트로이 목마(trojan horse)에 취약하다 (Figure 1)[4].



(Figure 1) 트로이 목마

하위등급 사용자가 테이블을 생성 후 그 테이블의 write 권한을 상위등급 사용자에게 준다. 상위등급 사용자는 자신에게 주어진 권한을 만족하는 테이블에 대해 연산을 실행한다. 이 경우 상위등급 사용자가 "Program P"에 의해 다른 테이블에 write 행위를 하는 것과 동시에 하위등급 사용자로부터 받은 테이블에 write하게 된다. 결국 상위등급 데이터가 하위등급으로 유출되는 것이다.

2.2 MAC(Mandatory Access Control)

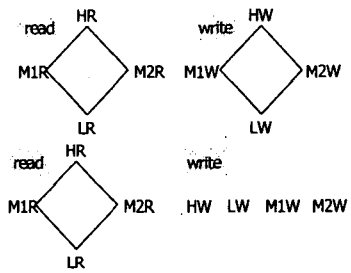
MAC은 비밀등급 비교에 의한 객체(object) 접근통제방식으로 대표적인 모델은 BLP(Bell-LaPadula) Model[5]을 들 수 있다. BLP Model은 각 보안등급 간에 상위등급의 정보가 하위등급으로 유출되는 것을 막기 위한 모델이다. 그 기본개념은 데이터베이스와 같은 객체(object)와 그 객체에 접근하는 일반적인 사용자로 표현할 수 있는 주체(subject), 객체에 할당된 등급(Classification), 주체에 할당된 등급인 접근허가(Clearance)로 나눈다. 객체와 주체간의 보안등급은 U(Unclassified), C(Classified), S(Secret), TS(Top-Secret)의 4단계로 구분한다. 보안등급은 U에서 TS로 올라 갈수록 높아지며 BLP Model은 두 가지 법칙을 갖는다.

- **Simple Security Property(no-read up)** : Subject S can read object O only if $L(S) \geq L(O)$
- ***(star)-Property(no-write down)** : Subject S can write object O only if $L(S) \leq L(O)$

하위등급의 주체는 상위등급 객체를 read 할 수 없으며, 상위등급 주체는 하위등급 객체에 write 할 수 없다. 만일 하위등급 주체가 상위등급에 write 할 수 없다면 하위등급은 상위등급에 자신의 등급으로 접근할 수 없는 정보가 있다는 것을 알 수 있으므로 하위등급으로 정보의 유출이 발생한다. BLP Model은 Polyinstantiation[6]을 허용하여 하위등급 주체가 상위등급 객체에 write 하는 것을 보장하며, 상위등급 주체와 하위등급 주체가 판별할 수 있는 정보를 구분해 하위등급은 마치 자신이 행한 write가 이상없이 수행되었음을 느낄 수 있게 한다. 하위등급 주체가 상위등급 객체를 보면 상위등급 객체는 null로 표현된다. 이것으로 하위등급은 상위등급의 정보를 유추할 수 없다. 또한, *-Property는 write-up이 적용되는 모델과 write-equal이 적용되는 두가지 형태로 구분한다[7].

- **Liberal *-Property** : Subject S can write object O only if $L(S) \leq L(O)$
- **Strict *-Property** : Subject S can write object O only if $L(S) = L(O)$

MAC 모델은 격자(Lattice) 모델이라 표현된다. LBAC(Lattice Based Access Control)은 read와 write를 구분해 처리한다. 이를 도식화해서 표현하면 다음과 같다(Figure 2).



(Figure 2) {Liberal and Strict} *-Property

2.3 RBAC(Role Based Access Control)

RBAC은 시스템에서 다수의 사용자(user)와 데이터 객체(data object), 인가(permission)를 관리하는 방법으로 사용된다[8]. RBAC은 LBAC과 마찬가지로 보안모델에서 주체와 객체간 read와 write 동작을 간단하게 도식화함으로써 관리를 용이하게 하며, Role은 Role name과 Privilege로 구성된다.

- **Privilege(x,m)** : x is the object, m is set of the access mode
- **Role(rname,rpset)** : rname is the name of role, rpset is set of privileges of the role

Role Graph는 방향성을 지닌 단선으로 Cycle을 형성하지 않는 Role Graph로 도식화한다.

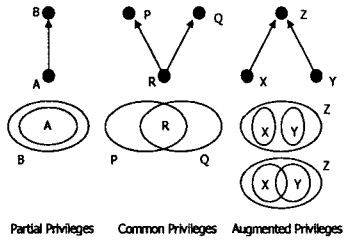
- **Partial Privileges** : $RoleA \subset RoleB$
- **Common Privileges** : RoleR is $RoleP \cap RoleQ$
- **Augmented Privileges** : $(RoleX \cup RoleY) \subset RoleZ$

Role Graph의 구성요소는 다음과 같다.

- **MinRole** : Minimum Privilege Set or $\varnothing$
- **MaxRole** : Set of every Privilege
- **junior** : RoleX is junior if $RoleX \rightarrow RoleY$
- **senior** : RoleY is senior if $RoleX \rightarrow RoleY$
- **path** : $RoleX \rightarrow RoleY$ if RoleX's Privilege Set $\subset$ RoleY's Privilege Set

• A path from MinRole to every Role
• A path from every Role to MaxRole

Role 관계는 is-junior 관계로 표현되는 Partial Privileges와 common-junior 관계로 표현되는 Common Privileges, common-senior로 표현되는 Augmented Privileges로 나눈다(Figure 3).



(Figure 3) Basic Role Relationships

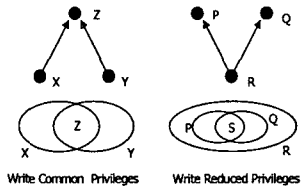
Role Graph의 장점은 대용량 데이터베이스 환경에서 이러한 Role Graph를 구현하고 그 형태를 유지, 재구성함으로써 관리자는 탄력성 있게 보안을 유지할 수 있다.

3. 역할관리(Role Administration) 보안모델

다중등급을 갖는 사용자(user)와 객체(object) 상호간의 인가(permission), 역할(role)을 관리하는 방법으로 확장된 MAC을 적용한 Role Graph를 제안한다. 시스템에서 권한 위임 측면을 살펴보면 권한을 가지고 있는 사용자가 권한이 없는 사용자에게 권한을 할당하는 동적 권한 위임과 관리자가 모든 사용자의 권한을 관리하는 정적 권한 위임으로 구분할 수 있다. 이러한 권한의 위임을 가장 잘 표현할 수 있는 것은 DAC이다. 그러나 DAC 기법은 각 사용자가 권한을 위임하는 행위로서 유용성이 크나 보안상 위험요소가 될 수 있다. 본 모델에서는 사용자 권한은 사용자 등급에 따라 판별하고 그 역할 관리는 보안 관리자가 직접 관리하는 것에 중점을 두는 형태로 중앙 집중적 보안관리에 초점을 두었다는 점이 Role Graph와의 차이점이라고 할 수 있다.

기존 LBAC 모델은 read와 write를 서로 분리해 역할계층을 표현한 후 병합(merge)한다. 이 경우 보안등급이 세분화되지 못하고 Unclassified, Secret, Top secret의 세가지 형태로 표현되었다. 또한 read와 write를 표현한 Role Graph 상에서 등급에 따른 권한 분류가 명확하지 못한 단점이 있다. 이것을 개선하기 위해 read의 경우 Simple Security Property를 적용하면서 Basic Role Relationship을 적용하였고, write의 경우 Liberal *-Property와 Strict *-Property를 적용하면서 Basic Role Relationship에 Extension Role Relationship(Figure 4)을 확장 적용했다. 실행(execute)은 같은 등급 내에서만 적용되는 것으로 제한한다.

이 방법을 사용한 Role Graph는 다중등급 객체가 각 등급별로 엄격하게 구분되고, 각 객체에 접근하는 주체도 자신의 등급에 맞는 등급에만 접근할 수 있다. 물론 read의 경우 상위등급 주체가 하위등급 객체에 접근할 수 있고, write의 경우는 주체의 등급 이상의 객체에 접근할 수 있음을 보장하여 보안의 누수를 막았다. 또한 관리자가 정책적으로 write의 경우 주체의 등급과 객체의 등급이 동일한 경우에만 write 할 수 있도록 정책을 조정할 수 있는 것을 보장한다. execute의 경우는 같은 등급 내에서만 실행되는 것을 보장하여 악의적인 스크립트의 실행과 같은 부당한 접근을 막는다.

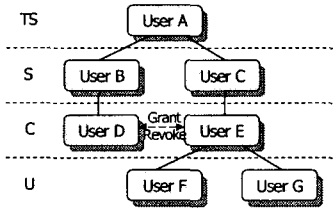


(Figure 4) Extension Role Relationships

• **Write Common Privileges** : RoleZ is RoleX∩RoleY
• **Write Reduced Privileges** : (RoleP is subset of RoleR) and (RoleQ is subset of RoleR) and ((RoleP∪RoleQ)⊂RoleR)
• **execute** : RoleA→RoleB only if RoleA's Level are equal to RoleB's Level
• **MaxRole(or MaxRole(read)∪MinRole(write))** : Privilege Set if defined(or all read Privilege∪∅)
• **MinRole(or MaxRole(write)∪MinRole(read))** : Privilege Set if defined(or all write Privilege∪∅)
• **(rname, read.rpset)** : RoleA→RoleB only if Basic Role Relationship
• **(rname, write.rpset)** : RoleA→RoleB only if Extension Role Relationship

4. 사례연구

각 등급 내에서 사용자의 역할 계층이 주어져있다(Figure 5). 각 사용자는 자신에게 할당된 등급 내에서 자신에게 할당된 테이블에만 접근한다. 자신의 등급과 동일한 등급의 테이블이라 할지라도 자신에게 할당되지 않은 테이블이라면 접근을 불허한다.



(Figure 5) 사용자의 역할계층 그래프

• **User [name]** : 사용자는 보안등급이 할당되어 있고, 각 사용자는 역할계층을 이루고 있다.
• **Table Secret Level** : 테이블 보안등급
• **Table [name] [Table Secret Level] operation(r/w)** : 보안등급이 있는 테이블은 각 사

용자에게 할당된 access를 제공한다.

• **execute(grant and revoke)** : 같은 등급 내에서만 각 사용자에게 할당된 테이블 권한 이동

각 테이블은 보안등급을 가지고 있고 주어진 operation을 허용하며 각 사용자에게 할당되어 있다(Table 1). 각 사용자는 자신에게 할당된 역할을 동일 등급 내에서 다른 사용자에게 할당하며(Alg 1) 회수할 수 있다(Alg 2).

User[name]	Table[name][Table Secret Level]operation(r/w)
User A	T1[TS] r, T2[TS] r/w
User B	T3[S] r, T4[S] w
User C	T5[S] r/w
User D	T6[C] r/w
User E	T7[C] r, T8[C] w
User F	T9[U] r/w
User G	T9[U] r/w

(Table 1) 사용자에게 할당된 테이블

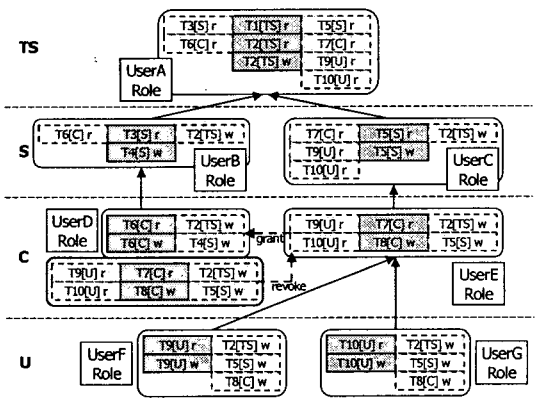
Grant ***Role[name]'s Privileges (Table[name] [Table Secret Level]operation(r/w))***
on ***User[name]'s Role only equal security level***
to ***User[name]***
with ***option*** ;

(Alg 1) 역할의 할당

Revoke ***Role[name]'s Privileges (Table[name] [Table Secret Level]operation(r/w))***
on ***User[name]'s Role***
from ***received User[name]'s Role*** ;

(Alg 2) 역할의 회수

각 사용자는 보안등급이 있고, 같은 보안등급 내에서도 각 역할은 구분되어 있다. UserE의 부재로 인한 역할의 승격은 UserD가 담당하며, UserE가 가지고 있는 모든 권한을 (Alg 1)에 의해 전수 받는다. 또한 UserD는 (Alg 2)에 의해 권한을 되돌릴 수 있다. UserE의 권한을 모두 전수 받아 권한을 실행할 때, 상위등급 데이터에 write하는 행위는 BLP 모델의 Polyinstantiation을 허용해 하위등급 사용자는 상위등급에 write 한 것으로 판단해 보안이 유지된다(Figure 6).



(Figure 6) Role Graph

5. 결론

본 논문에서는 주체와 객체의 관계를 Role Graph로 표현하는데 있어 read의 경우 RABC의 Basic Role Relationships를 적용하고 write의 경우 Extension Role Relationships을 적용하여 다중등급을 갖는 데이터베이스 환경에서의 권한과 역할을 표현했다. read의 경우는 상위등급 사용자로 갈수록 많은 역할을 갖게 되며, write의 경우는 하위등급 사용자로 갈수록 많은 역할을 갖게 된다. 기존의 Role Graph에서는 MaxRole의 경우는 모든 Role의 합으로 표현되고, MinRole의 경우는 최소의 집합 또는 ∅로 표현되었으나 Polyinstantiation을 허용하며 Simple Security Property와 *(star)-Property를 준수하기 위해 변형된 MAC 정책을 사용해 표현하였다. 또한 Liberal *-Property와 Strict *-Property 모두 Role Graph 상에 표현할 수 있음으로 관리자가 각 상황에 맞는 설계를 할 수 있도록 지원한다. 상위등급 정보에 하위등급 사용자가 write 함을 보장함으로써 하위등급 사용자는 상위등급 사용자가 사용하는 객체에 접근하지만 보안의 유출없이 정보를 보전할 수 있다.

6. 참고문헌

[1] D. Ferraiolo, J.F. Barkely, and D.R. Kuhn, Role Based Access Control:Features and Motivations, Annual Computer Security Applications Conference, 1995.

[2] R.S Sandhu. Latticed-based access control models. Computer, 26:9-19, Nov. 1993.

[3] Ravi S. Sandhu and Sushil Jajodia. Data and Database Securtiy and Controls. Handbook of Information Security Management, Auerbach Publishers, pp.481-499, 1993.

[4] Lewis, S.; Wiseman, S. Securing an object relational database. Computer Security Applications Conference, 1997.

[5] D.E. Bell and L.J. La Padula. Secure Computer System: Unified Exposition & Multics Interpretation. Technical report, Technical Report MTIS AD-A023588, MITRE Corporation, 1975.

[6] Teresa F. Lunt, Dorthy E. Denning, Roger R. Schell, Mark Heckman, and William R. Shockley, The SeaView Security Model, IEEE Transation On Software Engineering, Vol. 16, No. 6, June 1990.

[7] R.S. Sandhu. Role hierarchies and constraints for lattice-based access controls. In Computer Security - ESORICS 96, LNCS1146, pp.65-79. Springer Verlag, 1996.

[8] R.S. Sandhu. E.J. coyne, H.L. Feinstein, and C.E. Youman. Role-based access control models. Computer, 29:38-47, Feb. 1996.