Biba 모델과 역할기반 접근제어 모델의 상호 연동

최 은 복, 이 형 효, 노 봉 남
전남대학교 전산학과

Interworking Biba Model and Role-Based Access Control Model

EunBok Choi, HyungHyo Lee, Bongnam Noh
Dept. of Computer Science, Chonnam National University

요 약

개방형 통신환경에서 접근제어의 목적은 컴퓨팅 자원 및 통신 정보자원 등을 불법적인 사용자로부터의 사용, 노출, 수정, 파괴와 같은 비합법적인 행위로부터 보호하는데 있다. 본 논문에서는 상업적인 환경에서 조직의 관련된 작업을 기반으로 하는 역할기반정책에 주체와 객체에 등급을 부여하여 정보의 무결성을 보장하는 Biba 모델과의 연관성을 논하였다. 또한 인가등급을 갖는 주체를 역할그래프에 동적으로 배정할 수 있는 역할배정규칙과 제약조건을 정의하였으며 객체의 보안등급 비교유무에 따른 read-write 역할을 세분하여 그래프로 도식화하였다.

1. 서 론

분산처리환경에서 컴퓨터 보안은 시스템에 존재하는 정보를 부적절한 사용자로부터 보호하는데 그 목적이 있다. 특히 정보 통신망에서 운영되는 정보를 이용하는 사용자의 환경이 동적으로 변화하는 현대의 네트워크 환경에서는 접근제어 정책을 어느 하나의 정책에 근거한 일괄적 정의로는 부적절하다. 따라서 실질적인 접근제어 시스템에 적용이 가능하도록 관련된 몇가지 정책들간의 연관성을 연구할 필요가 있다.

접근제어 정책에는 다음과 같이 크게 세가지 정책으로 나눈다. 자율적 접근제어정책은 접근을 요청한 주체가 객체에 대한 권한을 자율적으로 다른 주체에게 권한을 부여하거나 철회할 수 있음을 의미한다. 이 정책에는 접근제어행렬(Access Control Matrix), 접근제어리스트(Access Control List), 능력리스트(Capability List) 등이 있다[8].

강제적 접근제어 정책은 시스템 관리자에 의해 보안등급이 결정되는 정책으로 대표적인 모델로는 비밀성을 중요시하는 BLP(Bell-LaPadula) 모델과 정보의 무결성을 강조하는

Biba 모델이 있다.

역할기반 접근제어정책은 주체가 가지는 역할에 따라, 접근할 수 있는 정보가 결정되고 사용할 수 있는 정보의 한계가 결정된다. 그리고 역할과 역할에 대한 권한을 정적으로 부여하므로서 수많은 접근권한을 관리하는데 융통성을 제공한다. 또한 역할이 조직이나 환경에 따라 자연스럽게 생성되고 재구성될 수 있는 유연성을 갖는다.

상업적인 환경에서는 주체가 자율적으로 권한을 부여하고 철회할 수 있는 접근제어행렬이나 리스트, 주체와 객체에 등급을 부여하여 제어하는 다단계 정책보다는 조직에 관련된 작업을 기반으로 하는 역할에 주체를 배정하는 역할기반 접근제어정책 사용이 적절하다.

본 논문에서는 Biba 모델의 보안 특성과 역할기반 접근제어 모델과의 관련성을 논하였으며 상업적인 환경에서 적용되는 역할기반 정책에 Biba 모델을 적용하여 정보의 무결성을 보장받도록 하였다. 또한 Biba 모델의 특성을 만족하면서 인가등급을 갖는 주체를 역할그래프에 동적으로 배정할 수 있는 역할배정규칙과 제약조건을 정의하였으며 객체의 보안등급의 비교유무에 따른 read-write 역할을 세분하여 그래프로 도식화하였다.

2. Biba 모델

BLP모델은 권한을 갖지 않는 사용자에게 정보가 흘러가는 것을 예방하는 비밀성에 기반을 둔 모델이다. 이 모델은 정보의 비밀성은 보장하지만 등급이 낮은 주체가 등급이 높은 객체 정보를 쓸 수가 있어 정보의 무결성을 보장하지는 못한다. 이러한 단점을 보완하기 위해 Biba 모델이 제안되었다. 이 모델에서도 주체와 객체의 보안등급에 의해 정책이 수행된다. 특히 보안등급을 무결성 등급이라 한다. 이 무결성 등급은 크게 두 가지로 분류한다. 하나는 Crucial(C), Very Important (VI), Important(I)로 구분되는 보안등급이고 다른 하나는 범주의 집합이다. 보안등급은 C > VI > I의 관계를 형성하며 범주의 집합은 BLP모델과 마찬가지로 비계층 관계를 갖는다.

만약, 보안등급과 범주의 집합이 각각 $C_1 \geq C_2$ 과 $S_1 \supseteq S_2$의 관계를 가지면 무결성 등급 $L_1=(C_1,S_1)$는 $L_2=(C_2,S_2)$를 지배한다. 무결성 등급이 $L_1 \geq L_2$ 나 $L_2 \geq L_1$ 의 관계가 모두 아니면 이 두 등급은 비교불가능(Incomparable) 하다고 한다.

Biba모델은 하나의 보안정책을 사용하지 않고 정보의 무결성을 보장하기 위해서 해당 보안 환경에 맞는 여러가지 보안 정책을 사용한다[2].

□ 주체의 최소상한 정책(Low-watermark policy for subjects)

- 주체의 무결성 등급이 객체의 무결성 등급을 지배한다면 주체는 객체에 쓰기 연산을 수행할 수 있다 ⇨ $L(s) \geq L(o)$ ⇒ Write
- 주체는 어떤 객체에 대해서도 읽기 연산을 수행할 수 있다.
  ⇨ whatever object ⇒ Read <단, L(s) = LUB[L(s), L(o)] >

하지만 이 연산을 부여한 후, 주체의 무결성 등급은 접근 되기 전의 주체와 객체의 무

결성 등급의 최소상한(Least Upper Bound)으로 등급을 고정시킨다. 만약, 주체 무결성 등급이 객체의 무결성 등급보다 높다면 read 연산을 수행한 후 주체의 등급이 객체의 등급에 맞춰져 낮아지고, 그렇지 않다면 주체의 등급을 그대로 유지한다.

이 정책의 주요 단점은 접근요청의 제출 순서에 의존적이다. 만약, 자신보다 낮은 객체를 읽기 위해서는 자신의 등급이 객체의 등급에 맞춰져서 등급이 낮아지므로 자신의 객체에 대한 쓰기 연산을 수행할 수 없다는 단점이 있다.

□ 객체의 최대하한 정책(Low-watermark policy for objects)

- 주체는 어떤 무결성 등급을 가진 객체이더라도 쓰기 연산을 수행할 수 있다.
  ⇨ whatever object ⇒ Write <단, L(o) = GLB[L(s), L(o)] >

이 정책은 객체의 무결성 등급이 쓰기 연산이 수행된 후, 주체와 객체의 무결성 등급의 최대하한(Greatest Lower Bound)으로 등급을 고정시킨다. 만약 주체의 무결성 등급보다 높다면 객체의 등급을 주체의 인가등급으로 낮추고, 그렇지 않으면 객체의 등급을 그대로 유지한다. 이 정책의 단점은 등급이 낮은 주체가 높은 등급의 객체에 부당하게 정보를 변경하는 정보의 무결성에 위배된다는데 있다.

□ 감사추적 정책(Low-watermark integrity audit policy)

- 주체는 어떤 무결성 등급을 가진 객체이더라도 쓰기 연산을 수행할 수 있다.
  ⇨ whatever object ⇒ Write
  <단, $L(s) \le L(o)$ or NOT{$L(s) \ge L(o)$ and $L(s) \le L(o)$} ⇒ Audit trail >

단, 주체가 자신보다 높은 객체나 비교 불가능한 등급의 객체에 대해 쓰기를 수행할 때에는 감사기록 화일에 기록된다. 이 정책은 2.2와 달리 무결성 등급이 고정되어있다. 하지만 이 정책도 등급이 낮은 주체가 높은 등급의 객체를 부당하게 변경할 수 있어서 정보의 무결성을 보장하지는 못한다.

□ 링 정책(Ring policy)

이 정책은 주체와 객체의 무결성 등급이 고정되어 있으며

- 주체의 무결성 등급이 객체의 무결성 등급을 지배한다면 주체는 객체에 쓰기 연산을 수행할 수 있다. ⇨ $L(s) \ge L(o)$ ⇒ Write
- 주체는 어떤 객체에 대해서도 읽기 연산을 수행한다. ⇨ whatever object ⇒ Read

이 정책은 주체가 자신보다 높은 객체나 비교 불가능한 등급의 객체에 대해서 수정하는 연산은 예방할 수 있다. 하지만 읽기 연산에 대한 제약조건이 없어 '부당한 수정(improper modification)'이 간접적으로 발생할 수 있어 이에 대한 원천적인 예방은 할 수 없다. 여기에서 간접적인 방법의 예로는, 높은 등급의 주체가 낮은 등급의 객체를 읽어 자신의 등급에 객체를 쓸 경우 등급이 낮은 정보가 상위 등급으로 흘러갈 우려가 있다.

□ 엄격한 무결성 정책(Strict integrity policy)

- Simple integrity : 주체의 무결성 등급이 객체의 무결성 등급에 지배된다면 주체는 객체에 읽기 연산을 수행할 수 있다. ⇨ $L(s) \leq L(o)$ ⇒ Read
- Integrity *-property : 주체의 무결성 등급이 객체의 무결성 등급을 지배한다면 주체는 객체에 쓰기 연산을 수행할 수 있다. ⇨ $L(s) \geq L(o)$ ⇒ Write

이 정책이 일반적인 Biba 모델이라 볼 수 있다. 이 정책은 지금까지 문제가 되어왔던 '부당한 수정' 위협을 예방할 수 있다. 또한 높은 등급의 주체가 낮은 등급의 객체를 읽고자 할 경우 객체의 등급으로 로그인하여 객체를 읽도록 하므로서 최소권한 규칙과 정보의 무결성을 지킬 수 있다.

3. 역할기반 접근제어 정책

역할기반 접근제어정책의 주요 동기는 상업적인 측면의 보안 정책을 강화시키기 위한 것과 임의적 접근제어정책과 강제적 접근제어정책으로부터 융통성과 세부적인 접근제어를 강화시키는데 있다. 많은 기업이 사용자에게 정보의 소유권을 부여하지 않고 회사나 대리점에 정보의 변경이나 삭제, 첨가 등 연산의 소유권을 부여하고 있다 이러한 강제적 접근제어 정책은 다단계 보안정책을 구현하는데 미흡하다. 이에 반해 역할기반 접근제어 정책은 상업적인 환경을 지원하는 다양한 보안 정책을 강화시킬 수 있다. 단, 역할기반 접근제어 정책의 구성요소들은 관리자에 의해 조정된다.

접근제어 정책은 역할과 허가사항, 사용자와 역할, 그리고 역할과 역할의 관계와 같은 역할기반 접근제어 정책 구성요소를 포함하는데 이들 구성요소는 시스템관리자에 의해 직접적으로 구성되거나 위임을 통해 간접적으로 구성된다.

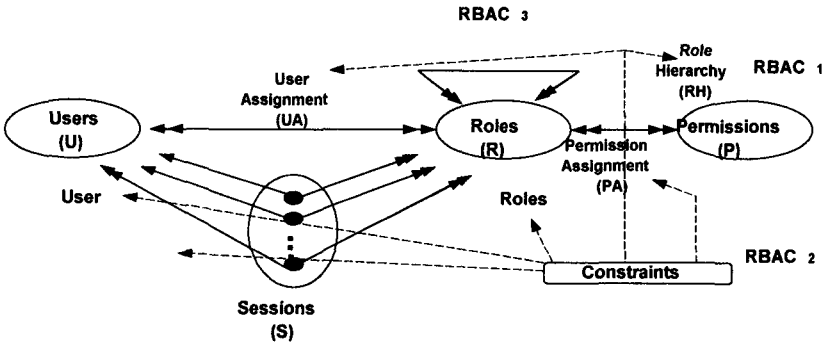
3.1 보안 원리

역할기반 접근제어 정책의 개념은 다음과 같이 잘 알려진 세 가지 보안 원리를 뒷받침한다[8].

- 최소권한원칙(least privilege principle) : 역할 계층성을 이용하여 작업에 꼭 필요한 최소한의 허가사항만을 역할에 배정하는 정책이다.
- 임무 분리(separation of duty) : 정보의 무결성을 침해하는 사기행위나 부정수단을 유발할 수 있는 작업은 상호 배타적인 역할로 지정하여 임무를 분리시켜 수행한다.
- 데이터 추상화(data abstraction) : 전형적인 운영체제나 시스템에서 사용되어졌던 데이터를 처리하는 read, write, execute 등의 허가사항 대신에, 다양한 기능을 수행할 수 있고 명령어를 추상화시키는 상업적인 처리 명령어 credit, debit, transfer, create account, delete account 등을 사용한다.

3.2 역할기반 접근제어 참조 모델

$RBAC_0$는 역할기반 접근제어 정책에서 최소한의 요구조건을 갖는 기본 모델이며 $RBAC_1$은 역할 계층성을 첨가한 개념이다. 반면 $RBAC_2$는 제약조건을 첨가한 모델이며 통합모델인 $RBAC_3$는 $RBAC_1$과 $RBAC_2$를 통합한 모델이다[6].



<그림 1> 역할기반 접근제어 모델

□ 기본 모델 - $RBAC_0$

$RBAC_0$는 4가지 개체인 사용자(U), 역할(R), 허가사항(P), 그리고 세션(S)으로 구성된다. 허가사항은 한 개 이상의 객체에 적용되는 접근 모드를 의미하며 이는 권한을 부여하는 양성적 측면을 가진다. 허가사항의 연산은 read, write, execute뿐만 아니라 상업적인 측면에서 추상적인 데이터를 처리할 수 있는 연산인 select, update, delete, debit, credit 등이 있다. 사용자는 역할을 수행하기 위해서는 트랜잭션에 해당하는 세션을 설정한다. 하나의 사용자가 여러 개의 역할을 동시에 수행할 수 있는 일 대 다 구조를 가질 수 있으며 사용자에 의해 자율적으로 생성, 변경, 소멸될 수 있다

□ 역할 계층성 모델 - $RBAC_1$

계층성은 권한과 책임을 수반하는 구조적 역할이라 할 수 있다. 권한은 단계별로 해당역할을 계층성에 부여하며, 역할에 대한 감사 추적시 계층구조를 이용한다. 하위역할은 상위역할에 모든 허가사항을 상속하므로 상위역할은 자신의 허가사항 뿐만 아니라 하위역할의 허가사항도 포함하게 된다. 때때로 상속성의 범위를 제한할 필요가 있는데, 하위역할이 상위 역할에 허가사항을 상속할 때 비밀을 요하는 경우에는 사설 비밀 역할(private Role)을 생성하여 상속한다.

□ 제약조건 모델 - $RBAC_2$

첫째, 가장 일반적인 RBAC의 제약조건은 상호 배타적인 역할(임무분리)이다. 예를 들어 회계관리자와 구매 관리자는 서로 배타적인 관계에 있으므로 임무를 분리시켜야 하는 제약조건을 두어야 한다. 둘째, 사용자가 가질 수 있는 역할의 개수(cardinality)를 제한할 수 있어야 한다. 이는 한 사용자가 수행할 수 있는 최대 역할개수와 최소 역할개수를 정

의함을 의미한다. 셋째, 조건의 역할로 한 사용자에게 꼭 필요한 역할은 선결조건으로 부여가 되어야 함을 의미한다. 기타 고려해야 할 사항으로 동적 임무 분리정책은 사용자와 역할이 설정은 되어있으나 세션이 수행될 때는 어느 하나만 수행되도록 해야한다는 것을 의미한다. 또한 세션 제약조건으로, 동시에 수행할 수 있는 사용자의 세션의 수를 제한해야 한다.

□ **통합 모델 - $RBAC_3$**

$RBAC_3$ 모델은 역할 계층성 모델인 $RBAC_1$과 제약조건 모델인 $RBAC_2$를 결합한 모델이다. 역할 계층 구조에 제약조건이 적용되며 역할 계층은 부분 순서 관계를 갖는다. 단일 시스템에 의한 역할기반 접근제어정책이 아닌 대규모 시스템에서는 역할의 개수가 매우 많아 이를 관리하는 일이 중요하다. 역할기반 접근제어정책의 주요한 장점은 이러한 허가사항 관리를 효율적이고 단순화시킬 수 있다는 점이다.

4. 역할그래프 모델

역할 그래프 모델은 사용자, 권한, 그리고 역할의 개념을 사용한다. 권한은 객체와 객체에 대한 연산의 집합인 (x, m)으로 구성되는데, x는 객체, m은 x에 대한 접근 모드를 나타낸다. 역할은 권한과 관련되어 명명된 집합인 (r.name, r.pset)으로 구성되며 r.name은 역할 이름, r.pset은 역할에 대한 권한집합을 의미한다. 역할은 역할 그래프의 한 노드를 구성하는데, 만약 R_1.rpset $\subseteq R_2$.rpset 이면 R_1이 R_2의 하위역할이 된다. 역할그래프는 다음과 같은 몇가지 특성을 갖는다[7].

- 하나의 최대역할을 가질 수 있다.
- 하나의 최소역할을 가질 수 있다.
- 역할 그래프는 비사이클을 형성해야 한다.
- 최소 역할에서 모든 r_i로 가는 경로가 존재한다.
- 모든 r_j에서 최대 역할로 가는 경로가 존재한다.
- 두 개의 역할 r_i 와 r_j에 대하여 만약 r_i.rpset $\subseteq$ r_j.rpset 이면 r_i에서 r_j로 가는 경로가 반드시 존재한다.

4.1 Biba 모델과 RBAC 모델의 상호연동

Biba 모델은 정보의 무결성을 보장하는 모델로 군사적인 환경에 적용이 가능한 BLP 모델과는 달리 상업적인 환경에 적합한 다단계 보안 모델이다. 그리고 역할기반 접근제어 모델은 역할과 역할에 대한 권한이 정적으로 기술된 상태에서 이 역할에 사용자를 동적으로 배정하므로써 수많은 접근권한을 관리하는데 융통성을 제공하며 역할이 조직이나 환경에 따라 자연스럽게 생성되고 재구성될 수 있는 유연성을 갖는다.

본 논문에서는 이러한 Biba 모델과 RBAC 모델의 상호 관련성을 알아보고 기존의 역할 그래프에 보안등급을 갖는 사용자를 배정하여 역할기반 모델에 정보의 무결성을 보장

하고자 한다. 역할 그래프에서는 한 주체가 역할에 배정되면 그 주체는 해당 역할의 모든 권한을 수행할 수 있을 뿐만 아니라 자신의 하위 역할의 모든 권한을 수행할 수 있게 된다.

□ **표기법 정의**

S : 주체의 집합 $S \in S$

R : 역할의 집합, $R \in R$

R_r : read only 역할

R_w : write only 역할

R_{rw} : read-write 역할

$\lambda(S)$: 주체의 무결성 보안등급

RoleAssign[S , R] : 주체 S의 역할 R에 대한 배정함수

Dominate(a,b) : $a \geq b$ or $a \supseteq b$

Equal(a,b) : Dominate(a,b) AND Dominate(b,a)

St : Trust Subject

w-level(R) : w-scope(R)중에서 최대의 보안등급

r-level(R) : r-scope(R)중에서 최소의 보안등급

w-scope(R) : write 접근모드를 갖는 모든 객체의 권한 집합

r-scope(R) : read 접근모드를 갖는 모든 객체의 권한 집합

□ **제약조건**

역할 R의 권한집합에서 객체 (o, r)에 해당하는 read 권한을 갖는 모든 객체의 집합을 r-scope(R)라고 한다. 그리고 역할 R의 권한집합에서 객체 (o, w)에 해당하는 write 권한을 갖는 모든 객체의 집합을 w-scope(R)이라고 하자. 역할에는 서로 다른 보안등급을 갖는 객체를 포함한다. 그러므로 우리는 r-scope(R)에 존재하는 모든 객체 중에서 최소의 보안등급을 갖는 등급을 해당 역할 R의 r-level(R)이라 정의한다. 그리고 w-scope(R)에 존재하는 모든 객체 중에서 최대의 보안등급을 갖는 등급을 해당 역할 R의 w-level(R)이라 정의한다.

먼저 그림2와 같이 read-only 역할(R_r)을 고려하여 보자. 이 역할의 경우는 w-scope의 값이 공집합이다. 만약 r-scope의 모든 객체의 등급이 모두 같은 보안등급을 갖는 경우, 이 역할은 Biba 모델의 simple integrity 정책이 적용된다. 만약, r-scope의 모든 객체들이 서로 다른 보안등급을 가질 경우, 주체의 인가등급이 해당 역할의 최소 보안등급에 해당하는 r-level(R)에 지배되어야 한다. 만약 그렇지 않으면 해당 주체는 자신보다 등급이 낮은 객체를 읽게 되어 정보의 무결성을 보장받지 못하게 된다. 왜냐면, 자신보다 낮은 객체를 읽어 자신의 등급에 쓸 경우 낮은 정보가 상위 등급으로 흐를 수 있기 때문이다.
이의 내용을 제약조건으로 표현하면 다음과 같다.

[제약조건 1] $\forall S \in S, \forall R_r \in R$

$RoleAssign(S, R_r) \Rightarrow \lambda(S) \leq r\text{-}level(R_r)$

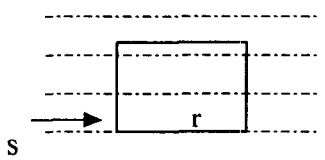
다음으로 그림3과 같이 write-only 역할(R_w)을 고려하여 보자. 이 역할의 경우는 r-scope의 값이 공집합이다.

만약 w-scope의 객체 등급이 모두 같은 보안등급을 갖는 경우, 이 역할은 Biba 모델의 integrity *-property 정책이 적용된다. 만약, w-scope의 모든 객체들이 서로 다른 보안등급을 가질 경우, 주체의 인가등급이 해당 역할의 최대 보안등급에 해당하는 w-level(R)을 지배하여야 한다. 만약 그렇지 않으면 주체가 자신보다 높은 등급이 객체를 쓰게 되어 정보의 무결성을 보장받지 못하게 된다.

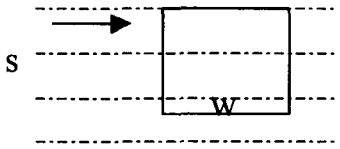
이의 내용을 제약조건으로 표현하면 다음과 같다.

[제약조건 2] $\forall S \in S, \forall R_w \in R$

$RoleAssign(S, R_w) \Rightarrow \lambda(S) \geq w\text{-}level(R_w)$



<그림 2> read-only 역할



<그림 3> write-only 역할

다음으로 그림4와 같이 read-write 역할(R_{rw})을 고려하자. 이 경우는 여러가지 경우의 가능성이 존재한다. 먼저 그림4의 역할 R1의 경우는 r-scope와 w-scope가 하나의 등급에 모두 존재하는 경우, 기존의 Biba 정책인 simple integrity 정책과 Integrity *-property를 적용한다.

그림4의 역할 R2과 같이 w-scope와 r-scope가 하나이상의 보안등급을 포함하면서 정확히 하나의 등급에 겹쳐있다면 이 역할은 겹쳐있는 등급을 갖는 객체에만 배정이 가능하다. 그러나 둘 이상의 등급이 겹쳐져 있다면 정보의 무결성을 해치므로 이 역할에는 배정이 불가능하다. 만약, 그림 4의 R3와 같이 r-scope와 w-scope가 겹치는 부분이 존재하지 않는다면 두 scope 사이의 등급에 해당하는 주체에 역할을 배정할 수 있다.

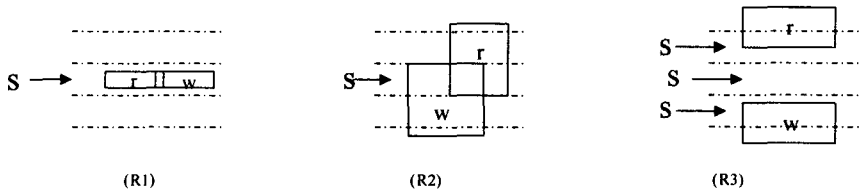
그림 4와 같이 r-scope가 w-scope보다 상위에 있는 경우에는 신뢰하는 주체나 비신뢰하는 주체든 상관하지 않고 해당 정책에 부합되는 주체에만 역할을 배정할 수 있다. 하지만 그림 5와 같이 w-scope가 r-scope보다 상위에 존재하는 경우에 신뢰할 수 없는 주체는 Biba 정책을 위반하는 read-down, write-up 경우가 발생하므로 배정할 수 없다. 단, 신뢰주체에는 Biba 정책의 Integrity *- propety를 적용하여, 만약 주체의 등급이 w-level(R)을 지배하면 해당 역할에 배정할 수 있다.

이와 같은 내용을 기반으로 제약조건 3을 정의하였다.

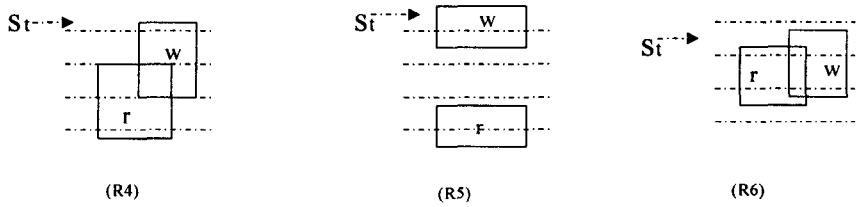
[제약조건 3] $\forall S \in S, \forall R_{rw} \in R$

RoleAssign(S, R_{rw}) ⇒

r-level(R_r) ≥ w-level(R_w) AND $\lambda(S) \leq$ r-level(R_r) AND $\lambda(S) \geq$ w-level(R_w)



<그림 4> read-write 역할



<그림 5> read-write 역할(신뢰 주체의 경우)

이들에 대한 역할배정규칙을 정의하면 다음과 같다.

□ **역할 배정 규칙**

[read-only 역할의 경우]

RoleAssign[S, R_r] =

True : if Dominate[r-level(R_r), λ(S)]

False : Otherwise

[write-only(append) 역할의 경우]

RoleAssign[S, R_w] =

True : if Dominate[λ(S), w-level(R_w)]

False : Otherwise

[read-write 역할의 경우]

RoleAssign[S, R_{rw}] =

True : if [Dominate(r-level(R_r), w-level(R_w)]

{

if Equal[r-level(R_r), w-level(R_w)] == λ(S) OR

[{r-scope(R_r) $\cap$ w-scope(R_w)} == $\varnothing$ AND r-level(R_r) $\geq$ λ(S) $\geq$ w-level(R_w)]

else if [S $\in$ St] AND Dominate[St, w-level(R_w)]

False : Otherwise

4.2 역할 그래프 도식화

먼저 역할의 r-level과 w-level이 비교 가능한 보안등급을 가진 경우의 역할을 고려하자. 이때의 r-level은 r-scope 객체중에서 최소의 등급을 취하고 w-level은 w-scope 중에서 최대의 등급을 취한다. 역할 그래프의 정의에 의해 R_1의 r-scope는 R_2의 r-scope에 모두 포함이 된다.

read 역할의 경우, R_2의 r-level은 R_1의 r-level과 같은 등급을 가져야 한다. 그렇지 않으면 역할 그래프 정의의 상속 개념에 의해 read up 정책이 위배된다.

[제약조건 4] R_1--> R_2 일 때, read 역할의 경우

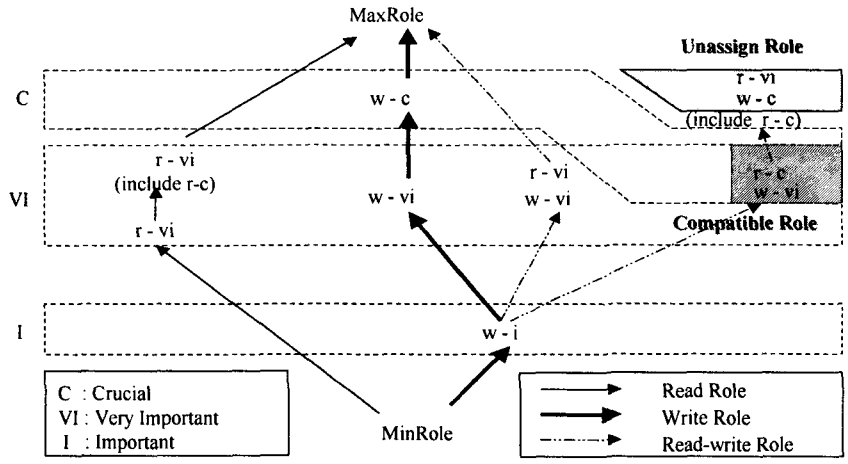
r-level(R_2) = r-level(R_1)

다음으로 write 역할의 경우 R_2의 w-level은 R_1의 w-level을 지배하여야 한다.

[제약조건 5] R_1 --> R_2 일 때, write 역할의 경우

w-level(R_2) $\geq$ w-level(R_1)

이들간의 역할 그래프 구조를 도식화하면 다음 그림6과 같다.



<그림 6> 비교 가능한 보안등급 역할 그래프 구조

다음으로 r-level과 w-level의 등급을 서로 비교할 수 없는 경우의 역할을 고려하자.

먼저 read 역할의 경우, r-level(R_2)는 자신의 하위 역할 R_1과 $R_{1'}$의 r-level중에서 최대 하한의 등급을 취한다. 이렇게 하므로써 read up 정책을 유지할 수 있다.

[제약조건 6] R_1 --> R_2, $R_{1'}$--> R_2일 때, read 역할의 경우

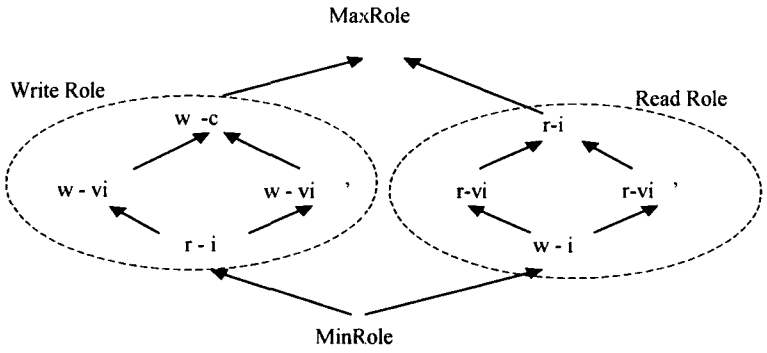
r-level(R_2) = GLB[r-level(R_1), r-level($R_{1'}$)]

다음으로 write 역할의 경우, w-level(R_2)는 자신의 하위역할 R_1과 $R_{1'}$의 w-level중에서 최소상한의 등급을 취한다. 따라서 write down 정책을 유지할 수 있다.

[제약조건 7] R_1 --> R_2, $R_{1'}$ --> R_2일 때, write 역할의 경우

w-level(R_2) = LUB[w-level(R_1), w-level($R_{1'}$)]

이들간의 역할 그래프를 도식화하면 다음 그림7과 같다.



<그림 7> 비교 불가능한 보안등급 역할 그래프 구조

5. 결 론

분산처리환경에서 컴퓨터 보안은 시스템에 존재하는 정보를 부적절한 사용자로부터 보호하는데 그 목적이 있다. 특히 정보 통신망에서 운영되는 정보를 이용하는 사용자의 환경이 동적으로 변화하는 현대의 네트워크 환경에서는 접근제어 정책을 어느 하나의 정책에 근거한 일괄적 정의로는 부적절하다. 따라서 실질적인 접근제어 시스템에 적용이 가능하도록 관련된 몇가지 정책들간의 연관성을 연구할 필요가 있다.

일반적인 역할 기반 정책에서는 주체나 프로세스에 해당하는 사용자, 권한과 책임이 수반되는 조직체의 작업기능에 해당하는 역할 그리고 역할이 수행하는 객체에 대한 권한, 즉 허가사항으로 구성된다. 상업적인 환경에 적용되는 역할기반 정책은 역할과 역할에 대한 권한이 정적으로 기술된 상태에서 이 역할에 사용자를 동적으로 배정하는 구조를 갖는다. 하지만 일반적인 분산환경에서 조직체의 관련된 작업기능을 수행하는 역할에 배정되는 주체나, 역할이 수행하는 허가사항에는 서로 다른 보안등급을 가진 수많은 객체들이 존재하게 된다.

본 논문에서는 Biba 모델의 보안 특성과 역할기반 접근제어 모델과의 관련성을 논하였으며 상업적인 환경에서 적용되는 역할기반 정책에 Biba 모델을 적용하여 정보의 무결성을 보장받도록 하였다. 또한 Biba 모델의 특성을 만족하면서 인가등급을 갖는 주체를 역할그래프에 동적으로 배정할 수 있는 역할배정규칙과 제약조건을 정의하였으며 객체의 보안등급의 비교유무에 따른 read-write 역할을 세분하여 그래프로 도식화하였다.

[참고문헌]

[1] Charles P.Pfleeger, Security in Computing, Prentice Hall
[2] Silvana Castano, DATABASE SECURITY, ADDISON-WESLEY
[3] Warwick Ford, Computer Communications Security, Prentice Hall

[4] Matunda Nyanchama, Sylvia Osborn, "Modeling Mandatory Access Control in Role-Based Security Systems, Database Security IX status and prospects, 8, 1995. pp. 129-144.

[5] David F. Ferraiolo, Janet A. Cugini, D. Richard Kuhn, "Role-Based Access Control(RBAC):Features and Motivations", COMPUTER SECURITY APPLICATIONS Conference, IEEE, 12, 1995, pp. 241-248.

[6] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, Charles E. Youman, "Role-Based Access Control Models", COMPUTER SOCIETY, IEEE, FEB. 1996, pp.38-47.

[7] Sylvia Osborn, "Mandatory Access Control and Role-Based Access Control Revisited", Second ACM Workshop on RBAC, 11. 1997, pp. 31-40.

[8] Ravi S. Sandhu and Pierangela Samarati, "Access Control : Principles and Practice", IEEE Communications Magazine, 9, 1994, pp. 40-48.

[9] Ravi S. Sandhu, "Lattice-Based Access Control Models", IEEE COMPUTER, 11, 1993. pp.9-19.

[10] Ravi Sandhu, "Access Control : The Neglected Frontier", Proc. First Australasian Conference on Information Security and Privacy, 6. 1996.

[11] Ravi S. Sandhu, "Role Hierarchies and Constraints for Lattice-Based Access Controls", Proc. Forth European Symposium on Research in COMPUTER SECURITY, 9. 1996. pp. 1-19.