다단계 보안통제가 가능한 확장된 역할기반 접근통제 모델

김 학 범† · 홍 기 융†† · 김 동 규†††

요 약

역할기반 접근통제는 사용자의 역할에 기반을 둔 접근통제 방법으로 기존의 임의적 접근통제나 강제적 접근통제에 비하여 다양한 컴퓨터 · 네트워크 보안분야에 있어서 유연성과 적용성을 제공한다. 본 논문에서는 Ravi S. Sandhu가 최초로 제안한 역할기반 접근통제 기본 모델인 $RBAC_0$ 모델에 주체 및 객체의 역할과 역할에 대한 보안등급을 추가로 고려한 확장된 역할기반 접근통제($ERBAC_0$ *Extended* $RBAC_0$) 모델을 새롭게 제안하였다. 제안된 $ERBAC_0$ 모델은 기존의 $RBAC_0$ 모델에 비하여 주체 및 객체 수준에서 다단계 보안통제 및 역할기반 접근통제를 보다 정교하게 제공할 수 있다.

Extended Role Based Access Control Model with Multilevel Security Control

Hak-Beom Kim† · Ki-Yoong Hong†† · Dong-Kyoo Kim†††

ABSTRACT

RBAC(*Role Based Access Control*) is an access control method based on the user's roles and it provides more flexibility and applicability on the various computer and network security fields than DAC(Discretionary Access Control) or MAC(Mandatory Access Control).

In this paper, we newly propose $ERBAC_0$(*Extended* $RBAC_0$) model by considering subject's and object's roles and security levels for roles additionally to $RBAC_0$ model which is firstly proposed by Ravi S. Sandhu as a base model. The proposed $ERBAC_0$ model provides finer grained access control with multilevel security on the base of subject and object level than $RBAC_0$ model.

1. 서 론

역할기반 접근통제(*RBAC* : *Role Based Access Control*) 개념은 1970년대 다중사용자와 다중 응용을 위한 온라인 시스템에서 시작되어 최근 들어 기존의 임의적 접근통제(DAC : Discretionary Access Control) 및 강제적 접근통제(MAC : Mandatory Access Control)에 비하여 정교하고 유연성을 제공함으로서 관심이 집중되고 있다[1]. 특히 의료시스템 분야[2], 웹 환경[3-7]과 데이터베이스 분야[8-10]에 역할기반 접근통제를 적용하는 활발한 연구가 진행되고 있다. 역할기반 접근통제의 주요 동기는 수행이 어려운 보안관리 과정을 관리자가 능률적으로 처리하고 국방분야 이외의 상용분야에 적합한 보안정책을 정확히 표현하고 적용할 수 있도록 하는데 있다[11, 12].

1993년 NIST(National Institute of Standards and Technology)는 정부와 산업체의 협동작업으로 상용, 정

† 정 회 원 한국정보보호센터 선임연구원
†† 종신회원 · (주)케이사인, (주)시큐브 대표이사
††† 정 회 원 아주대학교 컴퓨터공학과 교수
논문접수 2000년 1월 19일, 심사완료 · 2000년 6월 2일

부 및 국방 분야의 정보보호의 필요성에 대한 연구를 진행하여 다양한 응용 환경에 적용할 수 있는 새로운 접근통제 정책의 필요성을 인식하였다. 이러한 연구를 기반으로 현재까지 NIST를 중심으로 역할기반 접근통제 분야에 있어 독립적으로 3가지가 개발되어 진행중이다. 그 첫째가 조지메이슨 대학 및 Seta Corporation의 Ravi S. Sandhu 박사가 주축이 되어 진행되고 있는 역할기반 접근통제의 정의 및 실현가능성에 대한 연구가 있다. 두 번째가 NIST의 John Barkley를 중심으로 health care 시스템에 역할기반 접근통제를 적용하는 시도이며, 세번째가 메릴랜드 대학의 Virgil Gligor 박사를 중심으로 접근통제에 안전하고, 효과적이며 일관적인 메커니즘을 적용하기 위한 역할기반 접근통제에 대한 정형적 참조모델(Formal Reference Model) 개발이다. 이를 위하여 NSA(National Security Agency)와 공동연구개발을 통하여 개발된 역할기반 접근통제에 대한 정형적 참조 모델을 Mach 운영체제에 기반한 NSA의 Synergy 플랫폼 상에 구현 중에 있다[13]. 이런 노력의 결실로 역할기반 접근통제는 SESAME(Secure European System for Applications in a Multi-vendor Environment) 분산 시스템과 OMG(Object Management Group)의 CORBA(Common Object Request Broker Architecture) 보안명세서[14]에서도 분산 객체 기술로 사용할 수 있는 접근통제 메커니즘의 하나로서 사용토록 하고 있다. 역할기반 접근통제 관련 기술을 구현하고 있는 제품도 Oracle, Sybase, Lotus Notes, Microsoft Transaction Server 등 점차적으로 증가하고 있는 추세에 있다[15]. 이와 함께 '98년 5월 버전 2.0이 발표되어, 1999년 6월 ISO/IEC 15408 국제표준으로 제정된 국제공통평가기준(CC : Common Criteria)[16]에 역할기반 접근통제 내용이 포함되었으며, 이를 기반으로 접근통제 기능을 평가하기 위한 역할기반 접근통제 보호프로파일(PP : Protection Profile)[17]이 발표되었다. 또한 ISO/IEC 9075인 SQL 표준에도 RBAC이 포함되어 있다[18].

역할기반 접근통제 정책은 역할-허가, 사용자-역할, 역할-역할 관계와 같은 다양한 역할기반 접근통제 요소들로 구체화된다 이러한 역할 및 역할 관계에 기반하여 특정 사용자가 시스템내의 자원이나 데이터에 대한 접근을 허용할 것인지가 결정되는 것이다. 접근허가 결정에 필요한 역할기반 접근통제 요소들은 시스템 관리자에 의해 직접적으로 설정되거나 또는 시스템 관리자에 의해 위임된 적절한 관리적 역할(administrative role)에 의해 설정될 수도 있다[19].

강제적 접근통제는 사용자와 객체에게 부과된 보안 레이블을 기반으로 접근통제를 수행하며 임의적 접근통제는 설정된 허가, 거부 정책에 기반하여 객체에 대한 접근을 통제한다. 그러나 역할기반 접근통제는 강제적 접근통제, 임의적 접근통제 정책과 함께 사용될 수 있는 접근통제의 독자적인 요소이다. 역할기반 접근통제에 있어서 강제적인 부분은 각 사용자가 어떤 허가 또는 사용자가 역할에 할당되는가에 대한 선택권을 가지지 못한다는 것이고, 임의적인 부분은 각 사용자가 이러한 결정을 만들 수 있다는 것을 의미한다. 따라서 역할기반 접근통제는 자체로서는 정책중립적이지만 역할기반 접근통제의 특정한 설정에 따라 강제적인 요소를 가지게 되고 또한 임의적인 요소를 가질 수도 있게 된다

비록 역할기반 접근통제가 정책 중립적이지만 세가지 보안 정책을 제공하고 있다. 그 첫 번째가 특권의 최소화(least privilege)로서 이는 역할기반 접근통제가 역할에 할당된 사용자들에 의해 수행되는 작업들이 단지 설정된 것에 의해 허가된 것만 가능하게 지원된다. 두 번째로 의무의 분리(separation of duties)는 재정관리와 같은 민감한 작업을 수행하기 위해 상호 배타적인 역할을 보장했을 때 가능해진다. 마지막으로 데이터 추상화(data abstraction)는 운영체제에서 제공되는 읽고 쓰기 권한이라기보다는 계정에 대한 예금과 출금 같은 추상화된 허가 방법에 의해 제공된다. 그러나 역할기반 접근통제는 이러한 원리들의 응용을 강제화할 수는 없다. 하지만 보안 관리자는 역할기반 접근통제의 설정에 있어 이들을 위반하도록 할 수도 있다.

역할기반 접근통제는 다음과 같은 장점을 가진다[12].

첫째, 관리자에게 편리한 관리 능력을 제공한다. 전통적인 접근통제 메커니즘의 경우 사용자의 접근권한 관리는 매우 귀찮은 작업이지만 역할기반 접근통제의 경우 사용자의 자격과 책임에 따라 역할의 구성원으로 사용자를 지정하고 부여된 업무에 따라서 사용자를 역할의 구성원에서 제외하고 새롭게 추가하는 것이 쉽게 이루어질 수 있다. 역할기반 접근통제에서의 연산은 사용자 개인별로 특정 연산을 수행하도록 허가하는 것이 불가능하며 오로지 역할과의 관계를 통해 조직의 기능 변화에 따라 역할과 관련된 연산의 삭제 및 추가가 자유롭게 이루어질 수 있다

둘째, 통제하고자 하는 객체단위로 접근통제를 수행하는 기존의 방법(예, ACL, Capability List 등)과는 달리 관리자가 역할, 역할계층(hierarchy), 관계(relationship), 제약(constraint)의 정립을 통하여 사용자의 행동을 정적 또는 동적으로 규제할 수 있으므로 시스템 관리자에게 객체단위가 아닌 추상적인 트랜잭션의 개념으로 접근을 통제할 수 있다 따라서, 역할기반 접근통제는 업무를 수행하는 실제 환경에 자연스럽게 접목될 수 있다.

셋째, 역할기반 접근통제가 분산환경에서 사용되는 경우 역할기반 접근통제 관리자의 책임을 중앙과 국지 보호 영역으로 구분할 수 있다.

기존의 역할기반 접근통제 모델은 역할이나 허가 등에 사용자만을 고려하고 있으므로 실제의 응용 시스템 상에서 정확한 접근통제를 위해서는 주체 및 객체를 추가로 고려할 필요성이 있으며 또한, 역할에 대한 보안레이블을 고려하여 다단계 보안통제가 가능한 역할기반 접근통제 모델을 고려해 볼 필요가 있다. 이를 위하여, 본 논문에서는 Ravi S. Sandhu가 최초로 제안한 역할기반 접근통제 기본 모델인 $RBAC_0$ 모델[20]에 주체 및 객체의 역할과 역할에 대한 보안등급을 추가로 고려하여 확장된 역할기반 접근통제($ERBAC_0$: *Extended* $RBAC_0$) 모델을 새롭게 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안한 모델을 표현하기 위한 용어를 정의하며, 3장에서는 역할기반 접근통제 기본 모델을 살펴본다. 4장에서는 확장된 역할기반 접근통제($ERBAC_0$: *Extended* $RBAC_0$) 모델을 새롭게 제안하고, 모델의 검증, 적용예와 함께 기존 모델과의 특성을 비교한다. 마지막으로 5장에서 결론 및 향후연구 과제를 기술한다.

2. 용어 정의

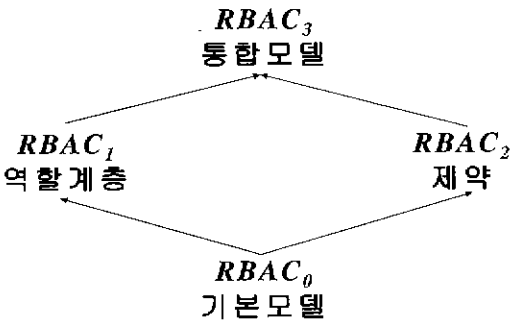
본 논문에서는 역할기반 접근통제 모델을 표현하기 위하여 다음과 같은 용어를 사용한다

- U : U는 사용자의 집합을 의미한다.
- C C는 세션의 집합을 의미한다.
- S : S는 주체의 집합을 의미한다. 여기에서 주체의 집합 S는 사용자의 집합 U를 포함하는 것으로 정의한다.
- S_{ui} : S_{ui}는 세션 c_i 및 사용자 u_i와 연관된 주체의 집합을 의미한다
- R . R은 역할(*Role*)의 집합을 의미한다
- R_{ui} : R_{ui}는 세션 c_i 및 사용자 u_i와 연관된 역할의 집합을 의미한다.
- SL : SL은 보안등급(*Security Level*)의 집합을 의미한다
- SL_u : SL_u은 사용자 u_i에게 부여된 보안등급의 집합을 의미한다.
- SL_o · SL_o은 객체 o에게 부여된 보안등급의 집합을 의미한다
- SL_r . SL_r은 역할 r에게 부여된 보안등급의 집합을 의미한다.
- CSL_{ui} : CSL_{ui}은 세션 c_i 및 사용자 u_i와 연관된 주체 S_{ui}에게 부여된 현재의 보안등급(*Current Security Level*)의 집합을 의미한다
- P : P는 허가(*Permission*)의 집합을 의미한다. 예를 들면, *READ* 또는 *WRITE* 등이 허가에 해당한다.
- P_{ri} . P_{ri}는 세션 c_i 및 사용자 u_i와 연관된 허가의 집합을 의미한다
- G : G는 신뢰된 주체의 집합을 의미한다. 여기에서 신뢰된 주체의 집합 G는 주체의 집합 S에 포함되는 것으로 정의한다.
- O : O는 객체의 집합을 의미한다. 여기에서 객체의 집합 O는 주체의 집합 S를 포함하는 것으로 정의한다.
- PA PA는 허가와 역할의 연관관계 집합을 의미한다.
- UA : UA는 사용자와 역할의 연관관계 집합을 의미한다.
- SA : SA는 주체와 역할의 연관관계 집합을 의미한다.
- OA : OA는 객체와 역할의 연관관계 집합을 의미한다
- USL · USL는 사용자와 보안등급의 연관관계 집합을 의미한다.
- SSL : SSL는 주체와 보안등급의 연관관계 집합을 의미한다.
- OSL : OSL는 객체와 보안등급의 연관관계 집합을 의미한다.
- RSL . RSL는 역할과 보안등급의 연관관계 집합을 의미한다

3. 역할기반 접근통제 기본 모델

3.1 Ravi S. Sandhu의 역할기반 접근통제 모델

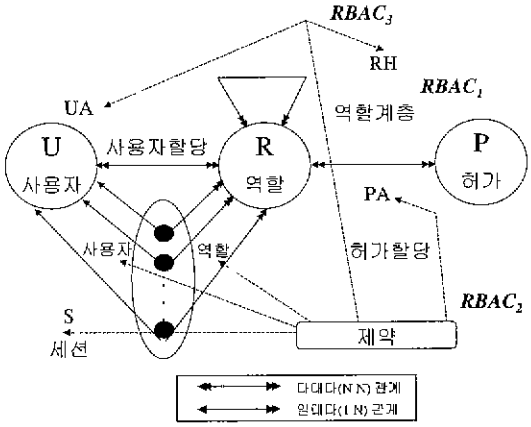
Ravi S Sandhu는 4가지 형태의 역할기반 접근통제 모델 $RBAC_0$, $RBAC_1$, $RBAC_2$, $RBAC_3$을 제안하였다[20] $RBAC_0$ 모델은 역할기반 접근통제를 다양한 시스템에 적용할 수 있도록 개발된 기본 모델이다. $RBAC_1$과 $RBAC_2$는 $RBAC_0$을 포함하지만 각각 고유한 특성을 보유하고 있다. $RBAC_1$은 다른 역할로부터 허가를 상속받을 수 있다는 역할 계층(role hierarchies)의 특성을 추가하였으며, $RBAC_2$는 역할기반 접근통제 요소들의 설정에 제한조건을 설정할 수 있도록 제약(constraints)을 가하는 특성을 부가하였다. 제약은 상위 레벨의 조직 정책을 결정하기 위한 강력한 메커니즘이다. 이 메커니즘에서는 어떤 역할이 상호 배타적으로 정해지면 역할에 대한 사용자의 할당은 조직의 전체적인 정책에 대한 손상을 고려할 필요 없이 분산시키거나 위임시킬 수 있다. 역할기반 접근통제의 관리가 완전히 보안 관리자에 의해 집중화되어 있을 경우 제약은 유용하게 활용될 수 있다. $RBAC_1$과 $RBAC_2$의 특징을 통합한 모델은 $RBAC_3$으로 (그림 1)에서와 같이 $RBAC_0$, $RBAC_1$과 $RBAC_2$를 수용한다. (그림 2)는 이 4가지 모델에 대한 개념도를 나타낸 것이다. 이들 중에서 $RBAC_0$ 모델은 $RBAC_1$, $RBAC_2$, $RBAC_3$ 모델들이 참조하는 기본 모델로 다음과 같이 정의된다.



(그림 1) 역할기반 접근통제 모델간의 관계

【정의 1】 역할기반 접근통제 모델($RBAC_0$ 모델)

U, R, P, C에 대하여 $RBAC_0$ 모델은 다음의 구성요소로 구성된다.



(그림 2) 역할기반 접근통제 모델

- $PA \subseteq P \times R$,
- $UA \subseteq U \times R$,
- $user : C \rightarrow U$,
- $roles : C \rightarrow 2^R$

여기에서, $user$는 각 세션 c_i를 단일 사용자인 $user(c_i)$에게 사상(mapping)시키는 함수를 의미하며, $roles$는 각 세션 c_i를 임의의 역할 집합인 $roles(c_i)$에게 사상시키는 함수를 의미한다. 이때, $RBAC_0$ 모델에서는 다음의 관계가 성립한다.

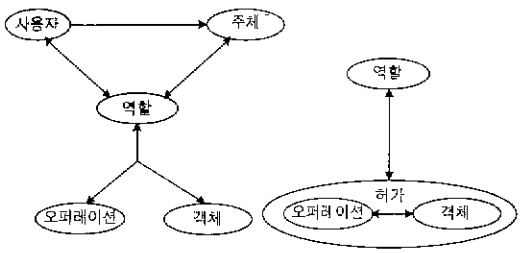
- $roles(c_i) \subseteq \{r | (user(c_i), r) \in UA\}$
- 세션 c_i는 $\bigcup_{r \in roles(c_i)} \{p | (p, r) \in PA\}$에 해당하는 허가를 지닌다.

3.2 W. A. Jansen의 역할기반 접근통제 모델

W. A. Jansen의 논문에서는 역할 계층의 특징과 이 특징들이 의무 분리(separation of duty)와 같은 기본적인 RBAC 특성에 어떠한 영향을 미치는지 분석하였다[21]. 기본적인 RBAC 모델의 특성은 역할의 멤버쉽에 대한 제약과 관련된 정적인 특성과 역할의 활성화(activation)에 대한 제약과 관련된 동적인 특성으로 구분된다. 이 논문에서는 두 가지 측면에서 역할 계층을 도입하였을 때 기본 특성들이 어떤 영향을 받는지를 분석하였다. RBAC의 기본 특성 중에서 역할 계층에 영향을 받는 특성에는 멤버의 최대 숫자를 제한할 수 있다는 cardinality의 상속성(inheritance), 의무 분리의 계층적 일관성(separation of duty hierarchical consis-

tency), 의무 분리의 상속성(separation of duty inheritance) 등이 있다.

이 논문에서 사용된 모델 구성요소와 객체를 고려한 허가는 (그림 3)과 같다. 이 모델에서의 정적인 특성과 동적인 특성은 다음과 같다. 여기에서, u는 사용자의 집합, x, y는 주체의 집합, i, j, k는 역할의 집합, op는 오퍼레이션, p, q는 허가를 나타낸다



(그림 3) (a) 모델 구성요소 (b) 객체를 고려한 허가

- 정적 Cardinality : $\forall i\ authorized - members[i] \leq membership - limit[i]$
- 정적 의무분리(SSD : Static Separation of Duty) :
  $\forall i \forall j \forall u\ i \in authorized - roles[u] \wedge j \in authorized - roles[u] \rightarrow \langle i, j \rangle \notin SSD$
- 정적인 운용상의 의무분리($SOSD$ Static Operational Separation of Duty) :
  $\forall i \forall j \forall u\ i \in authorized - roles[u] \wedge j \in authorized - roles[u] \rightarrow \langle i, j \rangle \notin SOSD$
- 동적 Cardinality :
  $\forall i\ active - members[i] \leq active - membership - limit[i]$
- 동적 의무분리(DSD : Dynamic Separation of Duty) .
  $\forall x \forall y \forall i \forall j\ i \in active - roles[x] \wedge j \in active - roles[y] \wedge active - user[x]\ active - user[y] \rightarrow \langle i, j \rangle \notin DSD$
- 동적인 운영상의 의무분리($DOSD$. Dynamic Operational Separation of Duty) .
  $\forall x \forall y \forall i \forall j\ i \in active - roles[x] \wedge j \in active - roles[y] \wedge active - user[x]\ active - user[y] \rightarrow \langle i, j \rangle \notin DOSD$

역할 계층의 개념을 제안한 모델에 적용하면 다음과 같은 영향을 미친다.

- 허용된 활동 :
  $\forall x \forall op \forall o\ exec[x, op, o] \equiv \exists i (i \in effective - roles[x] \wedge p \in authorized - permissions[i] \wedge \langle op, o \rangle \in p)$
- Cardinality 상속성(Cardinality Inheritance) :
  $\forall i \forall j\ \iota \geq \jmath \rightarrow (membership - limit[i] \leq membership - limit[j] \wedge (active - membership - limit[\iota] \leq active - membership - limit[\jmath])$
- 의무분리의 계층적 일관성(Separation of Duty Hierarchical Consistency) :
  $\forall \iota \forall j\ (i \geq j \wedge \exists k\ (k \geq i) \wedge (k \geq \jmath) \rightarrow \langle i, j \rangle \notin DSD \wedge \langle i, \jmath \rangle \notin SSD \wedge \langle i, \jmath \rangle \notin SOSD \wedge \langle i, j \rangle \notin DOSD)$
- 의무분리 상속성(Separation of Duty Inheritance) :
  $\forall \iota \forall j \forall k\ i \geq j,\ \langle j, k \rangle \in SSD \rightarrow \langle i, k \rangle \in SSD$
  $\forall i \forall \jmath \forall k\ i \geq \jmath,\ \langle \jmath, k \rangle \in DSD \rightarrow \langle i, k \rangle \in DSD$
  $\forall i \forall j \forall k\ i \geq j,\ \langle j, k \rangle \in SOSD \rightarrow \langle i, k \rangle \in SOSD$
  $\forall i \forall \jmath \forall k\ i \geq \jmath,\ \langle j, k \rangle \in DOSD \rightarrow \langle i, k \rangle \in DOSD$

3.3 NIST의 강화된 접근통제 모델

NIST에서는 RBAC의 특성에 기반을 두고서 프로토타입 구현, 시장 분석, Jansen의 연구에 대한 수정을 통하여 실제적으로 구현이 용이한 RBAC에 대한 강화된 모델을 정형적으로 제시하였다[22].

RBAC의 강화된 모델은 RBAC 권한 데이터베이스 모델인 MC_0와 RBAC 활성화 모델인 MC_1의 두 개의 구성요소(component)를 가진다. MC_0는 권한(authorization)에 대한 정적인 역할로의 보안 특성을 정의하였으며 MC_1은 역할의 동적 활성화에 대한 보안 특성을 정의하고 있다

MC_0(RBAC 권한 데이터베이스)는 다음과 같이 정의된다.

먼저 역할, 사용자, 허가(permission)에 대한 사상은 다음과 같이 정형화된다.

- 역할/멤버 사상(role/members mapping) :
  $RM(r : role) \rightarrow 2^{user}$
- 역할/허가 사상(role/permission mapping) :
  $RP(r : role) \rightarrow 2^{permissions}$

- 허가의 오퍼레이션에 대한 사상 :
  $POp(p : permission) \rightarrow \{operation\}$
- 허가의 객체에 대한 사상 :
  $POb(p : permission) \rightarrow \{object\}$

MC_0는 정적 특성을 정의하고 있으며 역할 계층, 상속, cardinality, 정적 의무분리 등이 포함된다.

- 역할 멤버쉽 상속(*Role Membership Inheritance*) :
  $(\forall i, j : role)(\forall u : user)\ i \geq j \wedge u \in RM[i] \Rightarrow u \in RM[j]$
- 정적 의무 분리(*SSD. Static Separation of Duty*) :
  $(\forall u : user)(\forall i, j : roles)\ u \in RM[i] \wedge u \in RM[j] \Rightarrow (i, j) \notin Ea$
  (여기에서 $Ea : role \times roles$는 배타적인 권한집합이다.)
- *Cardinality* : $(\forall r\ \ role) \# RM[r] \leq ML[r]$
  (여기에서 $ML(r : role) \rightarrow Z$는 역할 r에 대해 허가된 사용자의 수이다.)

정적 의무 분리와 Cardinality를 통해 다음과 같은 정적 특성이 유도될 수 있다.

【정리 1】 SSD와 익제의 일관성(Consistency of SSD and Containment)

$$(i, j) \in Ea \Rightarrow \neg(i \geq j \vee j \geq i)$$

【정리 2】 상호배타적인 역할의 비상속성(Non-Inheritance of Mutually Exclusive Roles)

$(i, j) \in Ea$이면 $k \geq i \wedge k \geq j$인 역할 k는 없다.

【정리 3】 상호배타의 이행성(Transitivity of Mutual Exclusion)

$$u \in RM[i] \wedge i \geq j \wedge (j, k) \in Ea \Rightarrow u \notin RM[k]$$

【정리 4】 정적 제약의 상속성(Inheritance of Static Constraints)

$i \geq j \wedge C(RM[j])$이면 $C(RU[i])$이다.
(여기서 C는 제약이다)

MC_1(RBAC 활성화 모델)은 다음과 같이 정의되는데, 동적 특성은 정적 특성을 보완하며 주체 또는 주체로부터 다른 기본 요소들로의 사상을 포함한다.

사용자, 주체, 역할에 대한 사상은 다음과 같이 정형화된다.

- $SU(s : subject) \rightarrow user$, 주체로부터 사용자로의 사상
- $AR(s : subject) \rightarrow 2^{role}$, 활성 역할(active role) 사상

MC_1은 동적 특성을 정의하고 있으며 역할 활성화, 동적 의무분리, 오퍼레이션 권한, 객체 접근 권한 등이 포함된다.

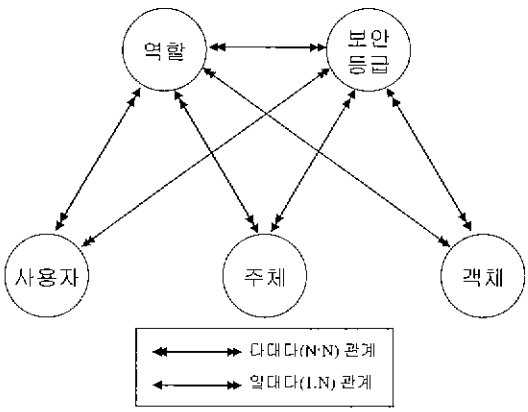
- 역할 활성화(*Role Activation*)
  $(\forall s\ \ subject,\ u : user,\ r : roles)\ \ r \in AR[s] \Rightarrow SU[s] \in RM[r]$
- 동적 의무 분리(*DSD : Dynamic Separation of Duty*) :
  $(\forall s,\ t : subject)(\forall i,\ j : role) : s \neq t\ \ i \in AR[s] \wedge j \in AR[t] \wedge (i, j) \in Er \Rightarrow SU[s] \neq SU[t]$
  (여기에서 $Er : role \times role$은 활성화시에 상호 배타적인 (i, j) 쌍이다.)
- 오퍼레이션 권한(*Operation Authorization*)
  $(\forall s : subject)(\forall op : operation)\ exec(s, op) \Rightarrow (\exists r : role)(\exists p : permission)\ r \in AR[s] \wedge p \in RP[r] \wedge op \in POp[p]$
  (여기에서 $exec(s, op)$는 주체 s가 오퍼레이션 op를 수행할 수 있으면 1, 그렇지 않으면 0이다.)
- 객체 접근 권한(Object Access Authorization)
  $(\forall s : subject)(\forall o : object)(\forall op : operation)\ access(s, op, o) \Rightarrow (\exists r : role)(\exists p : permission)\ r \in AR[s] \wedge p \in RP[r] \wedge op \in POp[p] \wedge o \in POb[p]$
  (여기에서 $access(s, op, o)$는 주체 s가 객체 o에 접근가능하면 1, 그렇지 않으면 0이다.)

4. 확장된 역할기반 접근통제 모델

4.1 모델 설계

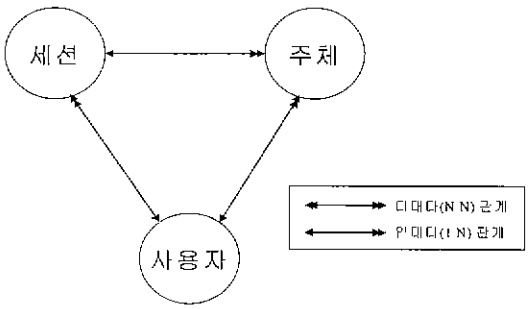
Ravi S. Sandhu가 제안한 $RBAC_0$ 모델은 사용자에 대해서만 고려하고 있으므로 주체 및 객체에 대하여

추가로 고려할 필요성이 있으며 또한, 역할에 대한 보안등급을 고려하여 다단계 보안통제를 가능케 함으로써 역할기반 접근통제 모델의 응용성을 확장시킬 필요성이 있다. W. A. Jansen이 제안한 역할기반 접근통제 모델에서는 주체 및 객체의 개념은 소개했으나 역할계층(Role Hierarchy) 문제를 해결하기 위한 것에 초점을 둔 것으로 객체의 역할, 사용자와 주체 및 객체에 대한 바인드는 정확히 해결하지 못하고 있는 상태이다. 따라서 본 절에서는 새로이 확장된 역할기반 접근통제($ERBAC_0$: *Extended RBAC*$_0$) 모델을 제안하기 위하여 먼저 사용자, 주체 및 객체와 역할간에 존재하는 관계성(Relationship)을 (그림 4) 및 (그림 5)와 같이 분석·제시하였다. (그림 4)는 사용자와 역할, 주체와 역할, 객체와 역할간의 관계성을 나타낸 것이다. 한 사용자는 여러 개의 역할을 할당받을 수 있고 하나의 동일한 역할에 대해서 여러 명의 사용자가 할당되어 있을 수 있으므로 사용자와 역할간의 관계는 다대다(N : N) 관계가 성립한다. 또한 이느 한 주체(또는 객체)는 여러 개의 역할을 할당받을 수 있고 하나의 동일한 역할에 대해서 다수의 주체(또는 객체)가 할당되어 있을 수 있으므로 주체(또는 객체)와 역할간의 관계는 다대다(N : N) 관계가 성립한다.



(그림 4) 사용자, 주체, 객체 및 보안등급과 역할간의 관계

다음 (그림 5)는 세션, 주체 및 사용자간의 관계성을 나타낸 것이다. 어느 한 사용자는 다수개의 세션을 유지할 수 있고 한 세션 동안에는 그 세션에 해당하는 단일의 사용자가 연관되어 있으므로 사용자와 세션간에는 일대다(1 : N)의 관계가 성립한다. 또한, 어느 한 사용자는 세션을 유지하는 동안 다수의 주체를 생성하여 원하는 작업을 실행하게 되며 어느 한 주체는 반드시 단일의 사용자에게 연관되어 있으므로 사용자와 주체간에는 일대다(1 : N)의 관계가 성립한다 따라서 어느 한 세션은 그 세션에 해당하는 주체가 생성하여 실행시킨 다수의 주체들과 연관되므로 세션과 주체간에도 일대다(1 N)의 관계가 성립한다



(그림 5) 세션, 주체 및 사용자간의 관계

【정의 2】 (그림 2) 및 (그림 4)와 같은 관계성으로부터 *U*, *R*, *P*, *S*, *O*, *SL*에 대하여 다음 관계를 정리한다

- $PA \subseteq P \times R$,
- $UA \subseteq U \times R$,
- $SA \subseteq S \times R$,
- $OA \subseteq O \times R$,
- $USL \subseteq U \times SL$,
- $SSL \subseteq S \times SL$,
- $OSL \subseteq O \times SL$,
- $RSL \subseteq R \times SL$.

확장된 역할기반 접근통제 모델을 설계하기 위하여 주체와 역할의 연관관계 집합인 *SA*, 객체와 역할의 연관관계 집합 *OA*, 사용자와 보안등급의 연관관계 집합인 *USL*, 주체의 보안등급의 연관관계 집합인 *SSL*, 객체와 보안등급의 연관관계 집합인 *OSL*, 역할과 보안등급의 연관관계 집합인 *RSL*을 새롭게 추가하였다.

【정의 3】 임의의 X에 대한 보안등급을 결정하는 함수는 다음과 같다.

$$f_{SL} : X \rightarrow SL.$$

여기에서 f_{SL}은 사용자, 객체, 또는 역할을 의미하는

X를 보안등급인 $f_{SL}(X)$에게 사상시키는 함수를 의미한다. 따라서 사용자 u_i, 객체 o, 역할 r에 대한 보안등급은 각각 $f_{SL}(u_i)$, $f_{SL}(o)$, $f_{SL}(r)$이 되며 다음의 식 (1)이 성립한다.

$$SL_u = f_{SL}(u_i),$$
$$SL_o = f_{SL}(o),$$
$$SL_r = f_{SL}(r) \cdots\cdots (1)$$

【정의 4】 임의의 세션과 연관된 사용자를 결정하는 함수 f_u는 다음과 같다.

$$f_u : C \rightarrow U.$$

여기에서, f_u는 각 세션 c_j를 단일의 사용자인 $f_u(c_j)$에게 사상시키는 함수를 의미하며 다음의 식 (2)가 성립한다.

$$u_i = f_u(c_j) \cdots\cdots (2)$$

【정의 5】 임의의 세션 및 사용자와 연관된 주체의 집합을 결정하는 함수 f_s는 다음과 같다.

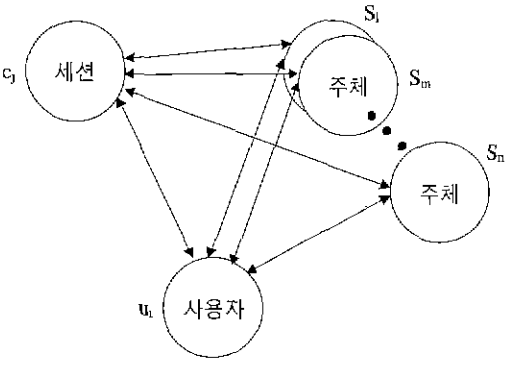
$$f_s : (C,\ U) \rightarrow S.$$

여기에서, f_s는 세션 c_j와 세션 c_j에게 연관된 특정의 사용자 u_i를 주체들의 집합인 $f_s(c_j,\ u_i)$에게 사상시키는 함수를 의미한다. 함수 f_u는 $RBAC_0$ 모델에서의 함수 $user$와 동일하며, 함수 f_s는 임의의 세션에 대하여 해당 사용자와 연관된 주체들의 집합을 결정할 수 있도록 하는 함수를 의미한다. 이 f_s는 $RBAC_0$ 모델에서는 정의되지 않는 함수로 확장된 역할기반 접근통제 모델을 설계하기 위하여 새롭게 추가하였다. 새롭게 추가된 함수 f_s는 함수 f_u를 이용하여 다음의 식 (3)과 같이 임의의 세션 c_j 및 사용자 u_i와 연관된 주체의 집합 S_{ji}을 얻어낼 수 있다.

$$S_{ji} = f_s(c_j,\ f_u(c_j)) = f_s(c_j,\ u_i) \cdots\cdots (3)$$

따라서, 사용자가 동작시킨 세션들은 무엇이며 어느 한 세션 동안에 실행한 주체들은 무엇인지를 파악할 수가 있다. 또한, 이러한 관계를 통하여 특정 사용자의 세션이 유지할 수 있는 역할을 결정할 수 있는 것이다. 이러한 관계는 (그림 6)에 나타낸 바와 같이 어느 한 사용자 u_i가 자신에게 속한 임의의 어느 한 세션 c_j 동안 원하는 작업을 수행하기 위하여 실행한 주체들은 s_l, s_m, …, s_n이며 세션 c_j가 가질 수 있는 역할은 사용자 u_i에게 부여된 역할과 이 사용자 u_i가 생성한 주체들 즉, s_l, s_m, …, s_n에게 부여된 역할에 의해서 결정된다. 즉, (그림 6)에서 임의의 세션 c_j 및 사용자 u_i와 연관된 주체의 집합 S_{ji}는 다음의 식 (4)와 같음을 알 수 있다.

$$S_{ji} = \{s_l,\ s_m,\ \cdots, s_n\} \cdots\cdots (4)$$



(그림 6) 세션-주체-사용자 바인드 관계

【정의 6】 세션 c_j 및 사용자 u_i와 연관된 주체 $s \in S_{ji}$의 현재 보안등급을 결정하는 함수 f_{CSL}은 다음과 같다.

$$f_{CSL} : (C,\ U) \rightarrow 2^{SL}.$$

여기에서, f_{CSL}은 세션 c_j 및 세션 c_j에게 연관된 특정의 사용자 u_i와 연관된 주체를 현재 보안등급의 집합인 $f_{CSL}(c_j,\ u_i)$에게 사상시키는 함수를 의미한다. 이 함수 f_{CSL}은 $RBAC_0$ 모델에서는 정의되지 않는 함수로 확장된 역할기반 접근통제 모델을 설계하기 위하여 새롭게 정의하였다. 새롭게 정의된 함수 f_{CSL}은 함수 f_u 및 f_s를 이용하여 다음의 식 (5)와 같이 임의의 세션 c_j 및 사용자 u_i와 연관된 주체 $s \in S_{ji}$에게 할당된 현재 보안등급을 얻어낼 수 있다.

$$\begin{aligned} CSL_{ji} &= f_{CSL}(c_j,\ u_i) \\ &\subseteq \bigcup_{s \in S_i} \{x \mid (s,\ x) \in SSL \wedge x \in SL_u\} \\ &= \bigcup_{s \subset fs(c_j,\ fu(c_j))} \{x \mid (s,\ x) \in SSL \wedge x \in SL_u\} \\ &= \bigcup_{s = fs(c_j,\ u_i)} \{x \mid (s,\ x) \in SSL \wedge x \in SL_u\} \end{aligned} \cdots\cdots (5)$$

이상의 정의에 따라 본 논문에서는 다단계 보안(MLS : Multilevel Security) 통제를 지원하는 역할기반 접근통제 모델을 설계하기 위하여 다음의 보안특성들을 새롭게 정의한다.

【MLS 조건 1】 *SRS(Simple Role Security)* 특성(property) : 역할에게 할당된 허가가 *READ*이고, 사용자의 보안등급이 역할의 보안등급보다 크거나 같은 경우에 사용자에게 역할이 부여될 수 있다. *SRS* 특성을 정형적으로 서술하면 다음과 같다.

$$SRS(u_i,\ r) = \begin{cases} TRUE, & where\ f_{SL}(u_i) \geq f_{SL}(r) \\ FALSE, & otherwise \end{cases}$$

【MLS 조건 2】 *SCRS(Simple Current Role Security)* 특성(property) : 역할에게 할당된 허가가 *READ*이고, 사용자의 보안등급이 세션 c_j 및 사용자 u_i와 연관된 주체 $s \in S_n$의 현재 보안등급보다 크거나 같고, 이 주체 s의 현재 보안등급이 역할의 보안등급보다 크거나 같은 경우에 주체에게 역할이 부여될 수 있다 *SCRS* 특성을 정형적으로 서술하면 다음과 같다.

$$SCRS(s,\ r) = \begin{cases} TRUE, & where\ f_{SL}(u_i) \geq f_{CSL}(c_j,\ u_i) \geq f_{SL}(r) \\ FALSE, & otherwise \end{cases}$$

【MLS 조건 3】 *CRS(Confinement Role Security)* 특성(property) : 역할에게 할당된 허가가 *WRITE*이고, 역할의 보안등급이 사용자의 보안등급보다 크거나 같은 경우에 사용자에게 역할이 부여될 수 있다. *CRS* 특성을 정형적으로 서술하면 다음과 같다.

$$CRS(u_i,\ r) = \begin{cases} TRUE, & where\ f_{SL}(r) \geq f_{SL}(u_i) \\ FALSE, & otherwise \end{cases}$$

【MLS 조건 4】 *CCRS(Confinement Current Role Security)* 특성(property) : 역할에게 할당된 허가가 *WRITE*이고, 사용자의 보안등급이 세션 c_j 및 사용자 u_i와 연관된 주체 $s \in S_n$의 현재 보안등급보다 크거나 같고, 이 주체 s의 현재 보안등급이 역할의 보안등급보다 크거나 같은 경우에 주체에게 역할이 부여될 수 있다 *CCRS* 특성을 정형적으로 서술하면 다음과 같다.

$$CCRS(s,\ r) = \begin{cases} TRUE, & where\ f_{SL}(r) \geq f_{CSL}(c_j,\ u_i) \geq f_{SL}(u_i) \\ FALSE, & otherwise \end{cases}$$

이상의 【정의 2~6】, 식 (1)~(5), 【MLS 조건 1~4】를 토대로 역할기반 접근통제를 위한 보안함수 및 오퍼레이션을 정의하면 다음과 같다.

【정의 7】 사용자 u_i에게 역할 r을 할당하는 것과 관련된 보안함수 및 오퍼레이션은 다음과 같다. 여기에서, $CanAssignR2U(u_i,\ r)$는 다음 3가지 경우를 판단하는 보안함수이다

① 역할 r에게 할당된 허가가 *READ*만 허용되어 있고 *SRS* 특성을 만족하는 경우에 사용자 u_i에 역할 r을 할당할 수 있는지 여부

② 역할 r에게 할당된 허가가 *WRITE*만 허용되어 있고 *CRS* 특성을 만족하는 경우에 사용자 u_i에 역할 r을 할당할 수 있는지 여부

③ 역할 r에게 할당된 허가가 *READ*와 *WRITE*가 모두 허용되어 있고 *SRS* 및 *CRS* 특성 모두를 만족하는 경우에 사용자 u_i에 역할 r을 할당할 수 있는지 여부

$AssignR2U(u_i,\ r)$는 $CanAssignR2U(u_i, r)$가 *TRUE*인 경우에 사용자 u_i에 역할 r을 할당하게 되므로 $(u_i,\ r)$을 집합 *UA*에 추가하는 기능을 수행하는 오퍼레이션을 의미한다. $RAssigned2U(u_i,\ r)$는 사용자 u_i에게 역할 r이 할당되어 있는지 여부를 판단하는 함수이다.

$$CanAssignR2U(u_i,\ r) = \begin{cases} TRUE, & where\ (READ,\ r) \in PA \\ & \quad \wedge (WRITE,\ r) \notin PA \wedge SRS(u_i,\ r) \\ TRUE, & where\ (READ,\ r) \notin PA \\ & \quad \wedge (WRITE,\ r) \in PA \\ & \quad \wedge CRS(u_i,\ r) \\ TRUE, & where\ (READ,\ r) \in PA \\ & \quad \wedge (WRITE,\ r) \in PA \\ & \quad \wedge SRS(u_i,\ r) \wedge CRS(u_i,\ r) \\ FALSE, & otherwise \end{cases}$$

$$AssignR2U(u_i,\ r) = \begin{cases} UA \leftarrow UA \cup (u_i,\ r), & where\ CanAssignR2U(u_i, r) \\ UA, & otherwise \end{cases}$$

$$RAssigned2U(u_i, r) = \begin{cases} TRUE, \text{ where } CanAssignR2U(u_i, r) \wedge (u_i, r) \in UA \\ FALSE, \text{ otherwise} \end{cases}$$

【정의 8】 주체 $s \in S_j$에게 역할 r을 할당하는 것과 관련된 보안함수 및 오퍼레이션은 다음과 같다 여기에서, $CanAssignR2S(s, r)$는 다음 3가지 경우를 판단하는 보안함수이다.

① 역할 r에게 할당된 허가가 *READ*만 허용되어 있고, 사용자 u_i와 역할 r이 *SRS* 특성을 만족하며 주체 s와 역할 r이 *SCRS* 특성을 만족하는 경우에 주체 s에 역할 r을 할당할 수 있는지 여부

② 역할 r에게 할당된 허가가 *WRITE*만 허용되어 있고, 사용자 u_i와 역할 r이 *CRS* 특성을 만족하며 주체 s와 역할 r이 *CCRS* 특성을 만족하는 경우에 주체 s에 역할 r을 할당할 수 있는지 여부

③ 역할 r에게 할당된 허가가 *READ*와 *WRITE*가 모두 허용되어 있고, 사용자 u_i와 역할 r이 *SRS* 및 *CRS* 특성 모두를 만족하며, 주체 s와 역할 r이 *SCRS* 및 *CCRS* 특성 모두를 만족하는 경우에 사용자 u_i에 역할 r을 할당할 수 있는지 여부

$AssignR2S(s, r)$은 $CanAssignR2S(s, r)$가 *TRUE*인 경우에 주체 s에 역할 r을 할당하게 되므로 (s, r)을 집합 SA에 추가하는 기능을 수행하는 오퍼레이션을 의미한다. $RAssigned2S(s, r)$는 주체 s에게 역할 r이 할당되어 있는지 여부를 판단하는 함수이다.

$$CanAssignR2S(s, r) = \begin{cases} TRUE, \text{ where } (READ, r) \in PA \wedge (WRITE, r) \notin PA \wedge CanAssignR2U(u_i, r) \wedge SCRS(s, r) \\ TRUE, \text{ where } (READ, r) \notin PA \wedge (WRITE, r) \in PA \wedge CanAssignR2U(u_i, r) \wedge CCRS(s, r) \\ TRUE, \text{ where } (READ, r) \in PA \wedge (WRITE, r) \in PA \wedge CanAssignR2U(u_i, r) \wedge SCRS(s, r) \wedge CanAssignR2U(u_i, r) \wedge CCRS(s, r) \\ FALSE, \text{ otherwise} \end{cases}$$

$$AssignR2S(s, r) = \begin{cases} SA \leftarrow SA \cup (s, r), \text{ where } CanAssignR2S(s, r) \\ SA, \text{ otherwise} \end{cases}$$

$$RAssigned2S(s, r) = \begin{cases} TRUE, \text{ where } CanAssignR2S(s, r) \wedge (s, r) \in SA \\ FALSE, \text{ otherwise} \end{cases}$$

이상의 【정의 2~8】, 식 (1)~(5), 【MLS 조건 1~4】를 토대로 세션 c_j 및 사용자 u_i와 연관된 역할 및 허가의 집합을 정의하면 다음과 같다

【정의 9】 세션 c_j 및 사용자 u_i와 연관된 역할의 집합을 결정하는 함수 f_r은 다음과 같다.

$$f_r : (C, U) \rightarrow 2^R.$$

여기에서, f_r은 세션 c_j와 세션 c_j에게 연관된 특정의 사용자 u_i를 역할의 집합인 $f_r(c_j, u_i)$에게 사상시키는 함수를 의미한다. 이 함수 f_r은 $RBAC_0$ 모델에서는 정의되지 않는 함수로 확장된 역할기반 접근통제 모델을 설계하기 위하여 새롭게 정의하였다. 새롭게 정의된 함수 f_r은 함수 f_u 및 f_s를 이용하여 다음의 식 (6)과 같이 임의의 세션 c_j 및 사용자 u_i와 연관된 역할의 집합 R_{ji}를 얻어낼 수 있다.

$$\begin{aligned} R_{ji} &= f_r(c_j, u_i) \\ &\subseteq \bigcup_{s \in S_n} \{r \mid RAssigned2U(u_i, r) \wedge RAssigned2S(s, r)\} \\ &= \bigcup_{s \in fs(c_j, fu(c_j))} \{r \mid RAssigned2U(u_i, r) \wedge RAssigned2S(s, r)\} \\ &= \bigcup_{s \in fs(c_j, u)} \{r \mid RAssigned2U(u_i, r) \wedge RAssigned2S(s, r)\} \quad \cdots\cdots (6) \end{aligned}$$

【정의 10】 세션 c_j와 세션 c_j에게 연관된 특정의 사용자 u_i에게 역할 r이 할당되어 있는지 판단하는 함수는 다음의 $RAssigned2UC(c_j, u_i, r)$와 같다.

$$RAssigned2UC(c_j, u_i, r) = \begin{cases} TRUE, \text{ where } r \in R_{ji} \\ FALSE, \text{ otherwise} \end{cases}$$

【정의 11】 세션 c_j 및 사용자 u_i와 연관된 허가의 집합을 결정하는 함수 f_p는 다음과 같다

$$f_p : (C, U) \rightarrow 2^P.$$

여기에서, f_p는 세션 c_j와 세션 c_j에게 연관된 특정의 사용자 u_i를 허가의 집합인 $f_p(c_j, u_i)$에게 사상시

키는 함수를 의미한다. 이 함수 f_p는 $RBAC_0$ 모델에서는 정의되지 않는 함수로 확장된 역할기반 접근통제 모델을 설계하기 위하여 새롭게 정의하였다. 새롭게 정의된 함수 f_p는 함수 f_r을 이용하여 다음의 식 (7)과 같이 임의의 세션 c_j 및 사용자 u_i와 연관된 허가의 집합 P_{ji}를 얻어낼 수 있다.

$$P_{ji} = f_p(c_j, u_i) = \bigcup_{r \in R_i} \{p | (p, r) \in PA\} \cdots (7)$$

이상의 【정의 2~11】, 식 (1)~(5), 【MLS 조건 1~4】를 토대로 Ravi S. Sandhu가 제안한 $RBAC_0$ 모델을 확장하여 다단계보안통제가 가능한 역할기반 접근통제($ERBAC_0$: *Extended* $RBAC_0$) 모델을 제안하면 다음 【정의 12】와 같다

【정의 12】 확장된 역할기반 접근통제 모델 : $ERBAC_0$ 모델

U, R, P, S, O, C, X에 대하여, $ERBAC_0$ 모델은 다음의 구성요소로 이루어진다.

- $PA \subseteq P \times R$,
- $UA \subseteq U \times R$,
- $SA \subseteq S \times R$,
- $OA \subseteq O \times R$,
- $USL \subseteq U \times SL$,
- $SSL \subseteq S \times SL$,
- $OSL \subseteq O \times SL$,
- $RSL \subseteq R \times SL$,
- $f_u : C \rightarrow U$,
- $f_s : (C, U) \rightarrow S$,
- $f_r : (C, U) \rightarrow 2^R$,
- $f_p : (C, U) \rightarrow 2^P$,
- $f_{SL} : X \rightarrow SL$,
- $f_{CSL} : (C, U) \rightarrow 2^{SL}$.

단, 여기에서 다음의 식 및 조건이 성립한다.

- $u_i = f_u(c_j)$,
- $S_{ji} = f_s(c_j, f_u(c_j)) = f_s(c_j, u_i)$,
- $R_{ji} = f_r(c_j, u_i)$
  $\subseteq \bigcup_{s \in S_{ji}} \{r | RAssigned2U(u_i, r) \wedge RAssigned2S(s, r)\}$,
- $P_{ji} = f_p(c_j, u_i) = \bigcup_{r \in fr(c_j, u_i)} \{p | (p, r) \in PA\}$.

4.2 $ERBAC_0$ 모델의 정형적 명세 및 검증

제안된 $ERBAC_0$ 모델을 검증하기 위하여, 이상의 【정의 2~11】, 식 (1)~(5), 【MLS 조건 1~4】를 토대로 $ERBAC_0$ 모델에 대한 정형적 명세와 검증(Formal Specification and Verification)을 제시하면 다음과 같다

【정의 13】 $ERBAC_0$ 시스템 $\sum$은 6-튜플 $<OP, R, V, v_0, F, Z>$이다. 여기에서 OP는 오퍼레이션의 집합 $\{AssignR2U, AssignR2S\}$, R은 역할의 집합, V는 시스템 상태의 집합, v_0는 시스템의 초기상태, F는 함수의 집합 $\{f_u, f_s, f_r, f_p, f_{SL}, f_{CSL}\}$, 그리고 Z는 시스템 전이함수(Transform Function)를 각각 나타낸다. 시스템의 전이함수 Z는 $OP \times V \rightarrow V$의 형태를 갖는 함수이다 단, 오퍼레이션 집합 OP는 적용되는 시스템에 따라 다양한 오퍼레이션을 포함할 수 있다

【정의 14】 $ERBAC_0$ 시스템의 상태 $v \in V$는 3-튜플 $<c_j, u_i, r>$이다

【정의 15】 $ERBAC_0$ 시스템의 상태 $v \in V$는 다음의 어느 한 조건이 만족하는 경우에 안전하다.

① 역할 r에게 할당된 허가가 $READ$만 허용되어 있고, 사용자 u_i와 역할 r이 SRS 특성을 만족하며 주체 s와 역할 r이 $SCRS$ 특성을 만족한다.
② 역할 r에게 할당된 허가가 $WRITE$만 허용되어 있고, 사용자 u_i와 역할 r이 CRS 특성을 만족하며 주체 s와 역할 r이 $CCRS$ 특성을 만족한다.
③ 역할 r에게 할당된 허가가 $READ$와 $WRITE$가 모두 허용되어 있고, 사용자 u_i와 역할 r이 SRS 및 CRS 특성 모두를 만족하며, 주체 s와 역할 r이 $SCRS$ 및 $CCRS$ 특성 모두를 만족한다

상기 조건을 정형적으로 서술하면 다음과 같다.

① $(READ, r) \in PA \wedge (WRITE, r) \notin PA \wedge SRS(u_i, r) \wedge SCRS(s, r)$
② $(READ, r) \notin PA \wedge (WRITE, r) \in PA \wedge CRS(u_i, r) \wedge CCRS(s, r)$
③ $(READ, r) \in PA \wedge (WRITE, r) \in PA \wedge SRS(u_i, r) \wedge SCRS(s, r) \wedge CRS(u_i, r)$

$\wedge CCRS(s, r)$

【정의 16】 $ERBAC_0$ 시스템의 실행내역 집합 Π는 $OP \times V$의 형태를 갖는 함수이다. 즉, 역할 $x \in OP$에 대하여 시스템의 실행내역 집합 Π는 다음과 같이 정의할 수 있다.

$\Pi(0) = \langle x, v_0 \rangle$, (시스템의 초기 실행내역)
$\Pi(1) = \langle x, v_1 \rangle \Rightarrow Z\langle x, v_0 \rangle = v_1$,
$\vdots$
$\Pi(n-1) = \langle x, v_{n-1} \rangle \Rightarrow Z\langle x, v_{n-2} \rangle = v_1$,
$\Pi(n) = \langle x, v_n \rangle \Rightarrow Z\langle x, v_{n-1} \rangle = v_n$.

【정의 17】 모든 실행내역이 안전한 경우에 $ERBAC_0$ 시스템 $\sum\langle OP, R, V, v_0, F, Z\rangle$은 안전하다.

상기 【정의 13~17】에 의하여 다음의 정리를 유도할 수 있다.

【정리】 $ERBAC_0$ 시스템 $\sum\langle OP, R, V, v_0, F, Z\rangle$이 다음 조건을 만족하는 경우에 안전하다.

① 시스템의 초기상태 v_0가 안전하다.

② $\forall x \in OP$, $\forall v \in V$, $\forall v^* \in V$에 대하여 $Z\langle x, v\rangle = v^*$의 관계가 성립하는 상태 v와 상태 v^*가 안전하다. 단, 여기에서 상태 v에서 임의의 오퍼레이션 x가 수행되면 다음의 상태 v^*로 전이됨을 의미한다.

【증명】 수학적 귀납법에 의하여 $ERBAC_0$ 시스템 $\sum\langle OP, R, V, v_0, F, Z\rangle$이 안전함을 증명하면 다음과 같다.

단계-1) 시스템의 초기상태 v_0 : $v_0 = \langle c_i^0, u_i^0, r\rangle$는 안전하다고 가정한다. 그러므로 【정의 7】의에서 정의한 $RAssigned2U(u_i^0, r)$가 $TRUE$이며, 【정의 8】에서 정의한 $RAssigned2S(s^0, r)$이 $TRUE$이다. 따라서, 【정의 15】에 의하여 다음 3가지중 어느 하나의 조건이 당연히 만족한다. 여기에서, $s^0 = f_s(c_j^0, u_i^0)$를 의미한다.

① $(READ, r) \in PA \wedge (WRITE, r) \notin PA$
$\wedge SRS(u_i^0, r) \wedge SCRS(s^0, r)$

② $(READ, r) \notin PA \wedge (WRITE, r) \in PA$
$\wedge CRS(u_i^0, r) \wedge CCRS(s^0, r)$

③ $(READ, r) \in PA \wedge (WRITE, r) \in PA$
$\wedge SRS(u_i^0, r) \wedge SCRS(s^0, r) \wedge CRS(u_i^0, r)$
$\wedge CCRS(s^0, r)$

단계-2) 상태 v_1 : $x \in OP$에 대하여 $Z\langle x, v_0\rangle = v_1$이 성립하며, $v_1 = \langle c_j^1, u_i^1, r\rangle$이라 한다. 여기에서, $s^0 = f_s(c_j^0, u_i^0)$ 및 $s^1 = f_s(c_j^1, u_i^1)$이 성립하며 다음의 2가지 상황이 고려될 수 있다.

① $x = AssignR2U(u_i^0, r)$인 경우 :

i) 역할 r에게 $READ$ 허가만 할당되어 있는 경우 .

【정의 7】에 의하여 $CanAssignR2U(u_i^0, r)$가 $TRUE$인 경우에 오퍼레이션 x가 정상적으로 실행되므로 $RAssigned2U(u_i^0, r)$가 $TRUE$이며 따라서, $SRS(u_i^0, r)$을 만족한다. 이때 그 결과로 $UA^1 \leftarrow UA^0 \cup \{(u_i^0, r)\}$가 성립하며 오퍼레이션 x가 실행된 이후에 u_i^0의 다음 상태인 u_i^1도 $SRS(u_i^1, r)$를 만족하게 된다.

$CanAssignR2U(u_i^0, r)$가 $FALSE$인 경우에는 $SRS(u_i^0, r)$를 만족하지 못하는 상태이므로 $RAssigned2U(u_i^0, r)$가 $FALSE$이다. 따라서, 그 결과로 $UA^1 \leftarrow UA^0$가 성립하여 변화가 없으므로 오퍼레이션 x가 실행된 이후에 u_i^0의 다음 상태인 u_i^1도 $SRS(u_i^1, r)$을 만족하게 된다.

따라서, 오퍼레이션 x가 실행된 이후에 u_i^0의 다음 상태인 u_i^1는 항상 $SRS(u_i^1, r)$을 만족한다.

ii) 역할 r에게 $WRITE$ 허가만 할당되어 있는 경우

【정의 7】에 의하여 $CanAssignR2U(u_i^0, r)$가 $TRUE$인 경우에 오퍼레이션 x가 정상적으로 실행되므로 $RAssigned2U(u_i^0, r)$가 $TRUE$이며 따라서, $CRS(u_i^0, r)$을 만족한다. 이때 그 결과로 $UA^1 \leftarrow UA^0 \cup \{(u_i^0, r)\}$가 성립하며 오퍼레이션 x가 실행된 이후에 u_i^0의 다음 상태인 u_i^1도 $CRS(u_i^1, r)$을 만족하게 된다.

$CanAssignR2U(u_i^0, r)$가 $FALSE$인 경우에는 $CRS(u_i^0, r)$를 만족하지 못하는 상태이므로 $RAssigned2U(u_i^0, r)$가 $FALSE$이다. 따라서, 그

결과로 $UA^1 \leftarrow UA^0$가 성립하여 변화가 없으므로 오퍼레이션 x가 실행된 이후에 u_i^0의 다음 상태인 u_i^1도 $CRS(u_i^1, r)$을 만족하게 된다.

따라서, 오퍼레이션 x가 실행된 이후에 u_i^0의 다음 상태인 u_i^1는 항상 $CRS(u_i^1, r)$을 만족한다.

iii) 역할 r에게 *READ*와 *WRITE* 허가가 할당되어 있는 경우 :

【정의 7】에 의하여 $CanAssignR2U(u_i^0, r)$가 *TRUE*인 경우에 오퍼레이션 x가 정상적으로 실행되므로 $RAssigned2U(u_i^0, r)$가 *TRUE*이며 따라서, $SRS(u_i^0, r)$과 $CRS(u_i^0, r)$을 모두 만족한다 이때 그 결과로 $UA^1 \leftarrow UA^0 \cup \{(u_i^0, r)\}$가 성립하며 오퍼레이션 x가 실행된 이후에 u_i^0의 다음 상태인 u_i^1도 $SRS(u_i^1, r)$과 $CRS(u_i^1, r)$을 모두 만족하게 된다.

$CanAssignR2U(u_i^0, r)$이 *FALSE*인 경우에는 $SRS(u_i^0, r)$ 또는 $CRS(u_i^0, r)$을 만족하지 못하는 상태이므로 $RAssigned2U(u_i^0, r)$가 *FALSE*이다. 따라서, 그 결과로 $UA^1 \leftarrow UA^0$가 성립하여 변화가 없으므로 오퍼레이션 x가 실행된 이후에 u_i^0의 다음 상태인 u_i^1도 $SRS(u_i^1, r)$과 $CRS(u_i^1, r)$을 모두 만족하게 된다.

따라서, 오퍼레이션 x가 실행된 이후에 u_i^0의 다음 상태인 u_i^1는 항상 $SRS(u_i^1, r)$과 $CRS(u_i^1, r)$을 모두 만족한다.

② $x = AssignR2S(s^0, r)$인 경우 :

i) 역할 r에게 *READ* 허가만 할당되어 있는 경우 :

【정의 7】에 의하여 $CanAssignR2S(s^0, r)$가 *TRUE*인 경우에 오퍼레이션 x가 정상적으로 실행되므로 $RAssigned2S(s^0, r)$가 *TRUE*이며 따라서, $SCRS(s^0, r)$을 만족한다. 이때 그 결과로 $SA^1 \leftarrow SA^0 \cup \{(s^0, r)\}$가 성립하며 오퍼레이션 x가 실행된 이후에 s^0의 다음 상태인 s^1도 $SCRS(u_i^1, r)$을 만족하게 된다.

$CanAssignR2S(s^0, r)$가 *FALSE*인 경우에는 $SCRS(s^0, r)$을 만족하지 못하는 상태이므로 $RAssigned2S(s^0, r)$가 *FALSE*이다. 따라서, 그 결과로 $SA^1 \leftarrow SA^0$가 성립하여 변화가 없으므로 오퍼레이션 x가 실행된 이후에 s^0의 다음 상태인 s^1도 $SCRS(s^1, r)$을 만족한다.

따라서, 오퍼레이션 x가 실행된 이후에 s^0의 다음 상태인 s^1는 항상 $SCRS(s^1, r)$을 만족한다.

ii) 역할 r에게 *WRITE* 허가만 할당되어 있는 경우 :

【정의 7】에 의하여 $CanAssignR2S(s^0, r)$가 *TRUE*인 경우에 오퍼레이션 x가 정상적으로 실행되므로 $RAssigned2S(s^0, r)$가 *TRUE*이며 따라서, $CCRS(s^0, r)$을 만족한다. 이때 그 결과로 $SA^1 \leftarrow SA^0 \cup \{(s^0, r)\}$가 성립하며 오퍼레이션 x가 실행된 이후에 s^0의 다음 상태인 s^1도 $CCRS(s^1, r)$을 만족하게 된다.

$CanAssignR2S(s^0, r)$가 *FALSE*인 경우에는 $CCRS(s^0, r)$를 만족하지 못하는 상태이므로 $RAssigned2S(s^0, r)$가 *FALSE*이다. 따라서, 그 결과로 $SA^1 \leftarrow SA^0$가 성립하게 되므로 오퍼레이션 x가 실행된 이후에 s^0의 다음 상태인 s^1도 $CCRS(s^1, r)$을 만족하게 된다.

따라서, 오퍼레이션 x가 실행된 이후에 s^0의 다음 상태인 s^1는 항상 $CCRS(s^1, r)$를 만족한다.

iii) 역할 r에게 *READ*와 *WRITE* 허가가 할당되어 있는 경우 :

【정의 7】에 의하여 $CanAssignR2S(s^0, r)$가 *TRUE*인 경우에 오퍼레이션 x가 정상적으로 실행되므로 $RAssigned2S(s^0, r)$가 *TRUE*이며 따라서 $SCRS(s^0, r)$와 $CCRS(s^0, r)$를 모두 만족한다 이때 그 결과로 $SA^1 \leftarrow SA^0 \cup \{(s^0, r)\}$가 성립하며 오퍼레이션 x가 실행된 이후에 s^0의 다음 상태인 s^1도 $SCRS(s^1, r)$와 $CCRS(s^1, r)$를 모두 만족하게 된다.

$CanAssignR2S(s^0, r)$가 *FALSE*인 경우에는 $SCRS(s^0, r)$ 또는 $CCRS(s^0, r)$을 만족하지 못하는 상태이므로 $RAssigned2S(s^0, r)$가 *FALSE*이다. 따라서, 그 결과로 $SA^1 \leftarrow SA^0$가 성립하여 변화가 없으므로 오퍼레이션 x가 실행된 이후에 s^0의 다음 상태인 s^1도 $SCRS(s^1, r)$와 $CCRS(s^1, r)$를 모두 만족하게 된다.

따라서, 오퍼레이션 x가 실행된 이후에 s^0의 다

올 상태인 s^1는 항상 $SCRS(s^1, r)$와 $CCRS(s^1, r)$을 모두 만족한다.

상기 ①과 ②에 의하여 v_1은 안전하다.

⋮

단계-n) 상태 v_n : $x \in OP$에 대하여 $Z\langle x, v_{n-1}\rangle = v_n$이 성립하며, $v_n = \langle c_j^n, u_i^n, r\rangle$이라 한다. 여기에서, $s^{n-1} = f_s(c_j^{n-1}, u_i^{n-1})$ 및 $s^n = f_s(c_j^n, u_i^n)$이 성립하며 다음의 2가지 상황이 고려될 수 있다.

① $x = AssignR2U(u_i^{n-1}, r)$인 경우 :
상기 단계 2)의 ①에서와 마찬가지로 오퍼레이션 x가 실행된 이후에 u_i^{n-1}의 다음 상태인 u_i^n는 항상 $SRS(u_i^n, r)$와 $CRS(u_i^n, r)$을 모두 만족한다.

② $x = AssignR2S(s^{n-1}, r)$인 경우 :
상기 단계 2)의 ②에서와 마찬가지로 오퍼레이션 x가 실행된 이후에 s^{n-1}의 다음 상태인 s^n는 항상 $SCRS(s^n, r)$와 $CCRS(s^n, r)$을 모두 만족한다.

상기 ①과 ②에 의하여 v_n은 안전하다

상기 단계-1), 단계-2), 단계-n)에 의하여 $ERBAC_0$ 시스템 $\sum\langle OP, R, V, v_0, F, Z\rangle$이 안전하다.

4.3 $ERBAC_0$ 모델의 적용 예

본 절에서는 키관리 시스템(Key Management System)을 위한 역할과 역할별 보안등급을 가정하여 제안된 $ERBAC_0$ 모델의 적용 예를 보인다. 다음의 <표 1>은 키관리 시스템에서 고려할 수 있는 역할과 역할에 부여할 수 있는 보안 등급의 예를 나타내고 있다.

<표 1> 키관리 시스템에서 고려할 수 있는 역할과 보안등급 예

역 할	내 용	보안등급
마스터키 생성 (MASTER_KEY_GEN)	암호키에 대한 최고비도용 마스터키 생성	Top Secret
고비도 암호키 생성 (HIGHLEVEL_KEY_GEN)	중요 데이터에 대한 고비도용 암호키 생성	Secret
일반 암호키 생성 (KEY_GEN)	일반데이터에 대한 중비도용 암호키 생성	Confidential
고비도 암호키 암호화 (HIGHLEVEL_KEY_ENC)	고비도용 암호키 암호화	Top Secret
일반암호키 암호화 (KEY_ENC)	중비도용 암호키 암호화	Secret

다음은 제안된 $ERBAC_0$ 모델이 키관리 시스템에서 동작할 수 있는 기본 알고리즘을 간략히 나타낸 것이다.

```
Procedure GrantSecurityLevelToRole(u_i, SL, role)
Begin
  if ( u_i is a Security Manager )
  begin
     r.type ← role ,          /* Assign type of role to
                                 variable role r */
     r.SecurityLevel ← SL ,   /* Assign security level of role
                                 to variable role r */

     switch(role) begin
     case MASTER_KEY_GEN :
          R ← R ∪ (WRITE, r)  /* Add role r to role set
                                 R */
          break;
     case HIGHLEVEL_KEY_GEN :
          R ← R ∪ (WRITE, r)  /* Add role r to role set
                                 R */
          break;
     case KEY_GEN :
          R ← R ∪ (WRITE, r)  /* Add role r to role set
                                 R */
          break;
     case HIGHLEVEL_KEY_ENC :
          R ← R ∪ (READ, r)   /* Add role r to role set
                                 R */
          break;
     case KEY_ENC :
          R ← R ∪ (READ, r)   /* Add role r to role set
                                 R */
          break;
     end switch
  end if
End

Procedure GrantSecurityLevelToUser(SL, u_i)
Begin
  if ( u_i is a Security Manager )
     begin
        u_i.SecurityLevel = SL ,   /* Assign security level
                                      SL to user u_i */
  end if
End

Procedure GrantRoleToUser(r, u_i)
Begin
  if ( u_i is a Security Manager )
     if( AssignR2U(u_i, r) )
        Print "Role" r "is successfully assigned to user"
        u_i ;
     else
        Print "Role" r "can not be assigned to user" u_i ;
  end if
End

Procedure GrantRoleToSubject(r, s )
Begin
   if( AssignR2S(s, r) )
      Print "Role" r "is successfully assigned to subject" s ;
   else
      Print "Role" r "can not be assigned to subject" s ;
End

Procedure KeyGen(u_i, c_j, r)
Begin
   if( RAssigned2UC(c_j, u_i, r) )
      Generate key subject to assigned role r ;
   else
      Print "user" u_i "can not generate key" ;
End
```

4.4 모델의 비교

역할기반 접근통제는 기본적으로 정보보호시스템 내에서 조직에서 정의된 임무나 또는 작업기능 등과 같은 역할에 기반하여 사용자의 자원에 대한 접근을 안전하게 통제하기 위한 수단을 제공하기 위한 것이다. 이때, 사용자에 대한 역할이 정의되있다고 하더라도 정보보호시스템 내에서 수행되는 사용자의 행위는 실행 가능한 프로그램(또는 객체), 즉 주체를 통하여 대신 이루어지게 된다. 그러므로 사용자는 정보보호시스템 내에서 허가된 임무를 수행하기 위하여 사용자를 대신하여 생성된 주체와 바인드(bind)가 필수적으로 요구된다 바인드에 필요한 정보는 사용자의 식별자 및 접근권한 등으로 역할 정보가 반드시 이에 포함되어야 한다 그러나 Ravi S. Sandhu가 제안한 기존의 $RBAC_0$ 모델은 이러한 주체 및 객체에 대한 역할 부여, 사용자와 주체 및 객체에 대한 바인드가 제공되지 않는다. 또한, W A. Jansen이 제안한 모델은 주체 및 객체의 개념은 소개했으나 기존의 $RBAC_0$ 모델이 지원하지 못하고 있는 역할 계층(Role Hierarchy) 문제를 해결하기 위한 주체의 역할에만 초점을 둔 것으로 이 모델 또한 객체의 역할, 사용자와 주체 및 객체에 대한 바인드가 제공되지 않고 있다. 그러므로 제안한 $ERBAC_0$ 모델을 사용할 경우 정보보호시스템은 사용자뿐만 아니라 주체 및 객체에 대한 역할을 별도로 지원할 수 있게 되며 또한, 사용자가 이들 주체 및 객체에 대하여 접근을 시도할 때 사용자의 역할과 주체 및 객체의 역할이 상호 바인드되어 정교한 역할기반 접근통제를 실현할 수 있게 되는 장점이 있다.

5. 결 론

역할기반 접근통제에 대한 연구는 '70년대초 다중 사용자와 다중 응용을 위한 상용 온라인 시스템에 적용해 보기 위한 것으로부터 시작되어 현재 기존의 접근통제 방법인 강제적 접근통제 및 임의적 접근통제에 이은 세 번째 접근통제 방법으로서 각광을 받고 있으며 특히 의료 분야, 웹 환경 및 데이터베이스 분야에 역할기반 접근통제를 적용하는 활발한 연구가 진행되고 있다. 향후에도 정형적 설계 및 검증, 시스템 실용화 등의 측면에서 많은 발전이 있을 것으로 기대된다.

〈표 2〉 모델 특징 비교

보안 모델 / 특징	Ravi S Sandhu 모델	W A Jansen 모델	NIST의 강화된 RBAC 모델	$ERBAC_0$ 모델
다단계 보안통제 기능	제공 안함	제공 안함	제공 안함	제공함
접근통제 대상	- 허가의 역할 - 사용자의 역할	- 허가와 역할 - 사용자의 역할 - 객체의 역할	- 허가의 역할 - 사용자와 역할 - 객체와 역할	- 허가의 역할 - 사용자와 역할 - 객체와 역할 - 주체와 역할 - 사용자의 보안등급 - 주체의 보안등급 - 객체와 보안등급 - 역할의 보안등급
역할 적용 대상	사용자	사용자, 주체	사용자 주체	사용자, 주체, 객체
장 점	- 기존의 DAC, MAC에 비하여 유연한 접근통제 정책 제공 - 사용자에 대한 역할 부여 기능	- 기존의 DAC, MAC에 비하여 유연한 접근통제 정책 제공 - 주체의 역할에 초점을 두어 역할 계층 문제를 해결	- DAC, MAC에 비하여 유연한 접근통제 정책 제공 - 직작의무분리, 최익피의 상속계층간의 관계 강화 - 주체의 역할에 기반한 오퍼레이션, 객체 접근 강화	- 기존의 DAC, MAC에 비하여 유연한 접근통제 정책 제공 - 주체 및 객체에 대한 역할 부여가 가능하므로 정교한 역할기반 접근통제 가능 - 사용자, 주체, 객체 및 역할에 대한 다단계 보안통제기 기능

본 논문에서는 Ravi S. Sandhu가 최초로 제안한 역할기반 접근통제 기본 모델인 $RBAC_0$을 개선하여 주체와 객체 및 역할에 대한 보안등급을 추가시켜 다단계 보안통제(Multilevel Security)를 가능케 하는 확장된 접근통제($ERBAC_0$: *Extended* $RBAC_0$) 모델을 새롭게 제안하였으며 설계된 모델에 대한 정형적 명세 및 검증, 적용 예를 제시하여 안전성 및 적용 가능성을 입증하였다. 본 논문에서 제안한 $ERBAC_0$ 모델은 기존의 Ravi S. Sandhu가 제안한 $RBAC_0$ 모델 및 W. A. Jansen 및 NIST에서 제안한 모델이 제공하지 못하는 주체 및 객체 수준에서의 역할기반 접근통제 기능과 보안등급에 기반한 다단계 보안통제 기능을 보다 정교하게 제공할 수 있다.

본 논문에서 제안한 $ERBAC_0$ 모델은 다양한 정보보호시스템에 적용하기 위해서는 역할 및 역할별 보안등급에 대한 고수준의 일반화(Generalization)가 요구되므로 이에 대한 지속적인 연구가 수행되어져야 할 것으로 사료된다.

참 고 문 헌

[1] David F. Ferraiolo and D. Richard Kuhn, "Role-based access controls," *15th NIST-NCSC National Computer Security Conference*, pp.554-563, Baltimore, MD, October 13-16, 1992.

[2] http : //hissa.ncsl.nist.gov/rbac/.

[3] John F Barkley, Anthony V. Cincotta, David F. Ferraiolo, Servan Gavrilla, and D Richard Kuhn, "Role Based Access Control for the World Wide Web," *20th NISSC National Information Systems Security Conference*, pp 331-340, Oct. 7-10, Baltimore Convention Center, Baltimore, MD, April 8, 1997.

[4] Larry S. Bartz, "hyperDRIVE : leveraging LDAP to implement RBAC on the Web," pp 69-74, *RBAC'97. Proceedings of the 2nd ACM workshop on Role-based access control*, Fairfax, VA, Nov. 6-7, 1997.

[5] David Ferraiolo and John Barkley, "Specifying and managing role-based access control within a corporate intranet," pp.77-82, *RBAC '97. Proceedings of the 2nd ACM workshop on Role-based access control*, Fairfax, VA, Nov 6-7, 1997.

[6] Ravi Sandhu and Joon S. Park, "Centralized user-role assignment for Web-based intranets," pp 1-12, *RBAC '98. Proceedings of the 3rd ACM workshop on Role-based access control*, Fairfax, VA, Oct. 22-23, 1998.

[7] Joon S. Park and Ravi Sandhu, "RBAC on the Web by Smart Certificates," *RBAC '99. Proceedings of the 4th ACM workshop on Role-based access control*, Fairfax, VA, Oct 28-29, 1999.

[8] Raymond K. Wong, "RBAC support in object-oriented role databases," pp.109-120, *RBAC '97. Proceedings of the 2nd ACM workshop on Role-based access control*, Fairfax, VA, Nov. 6-7, 1997.

[9] Tor Didriksen, "Rule based database access control - a practical approach," pp 143-151, *RBAC '97. Proceedings of the 2nd ACM workshop on Role-based access control*, Fairfax, VA, Nov. 6-7, 1997.

[10] Ravi Sandhu and Venkata Bhamidipati, "An Oracle implementation of the PRA97 model for permission-role assignment," pp.13-21, *RBAC'98. Proceedings of the 3rd ACM workshop on Role-based access control*, Fairfax, VA, Oct 22-23, 1998.

[11] David F Ferraiolo, Janet A. Cugini and D. Richard Kuhn, "Role-Based Access Control(RBAC) : Features and Motivations," *Annual Computer Security Applications Conference*, pp 554-563, IEEE Computer Society, 1995

[12] David Ferraiolo, Dennis M. Gilbert, and Nicklyn Lynch, "An examination of federal and commercial access control policy needs," *16th NIST-NCSC National Computer Security Conference*, pp 107-116, Baltimore, MD, September 20-23, 1993.

[13] Computer Systems Laboratory(CSL) Bulletin, "An Introduction to Role-Based Access Control," December, 1995.

[14] http : //csrc.nist.gov/rbac/sedan1.html, "Standard for Role-Based Access Control," 2000

[15] John F. Barkley, Konstantin Beznosov, and Jinny Uppal, "Supporting Relationships in Access Control Using Role Based Access Control," pp.55-65, *RBAC '99. Proceedings of the 4th ACM workshop on Role-based access control*, Fairfax, VA, Oct. 28-29, 1999.

[16] ISO/IEC 15408-2, *Information Technology - Security Techniques - Evaluation Criteria for IT Security - Part 2 : Security functional requirements*, Aug, 1999

[17] ISO/IEC 9075-2, *Information Technology - Database Language SQL - Part 2, Foundation(SQL : 1999)*, Nov, 1999.

[18] Jim Reynolds, Ramaswamy Chandramouli, *Role-Based Access Control Protection Profile*, Ver. 1.0, Cygnacom Solutions & NIST, July 30, 1998.

[19] Ravi Sandhu, "Role Hierarchies and Constraints for Lattice-Based Access Control," *Proc Fourth European Symposium on Research in Computer Security*, Rome, Italy, September 25-27, 1996

[20] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein and Charles E. Youman, "Role-Based Access Control Models," *IEEE Computer*, pp.38-47, Vol 29, No.2, February, 1996.

[21] W. A. Jansen, "Inheritance Properties of Role Hierarchies," *21th NCSC/NIST NISSC National Information Systems Security Conference*, pp 476-485, Crystal City, VA, October 5-8, 1998.

[22] David F. Ferraiolo, John F. Barkley, and D Richard Kuhn, "A Role Based Access Control Model and Reference Implementation within a Corporate Intranet," *ACM Transaction on Information System Security*, pp.34-64, Vol.2, No.1, Feb., 1999.

김 학 범

e-mail hbkim@kisa.or.kr

1990년 중앙대학교 대학원 전자계산학과(석사)

1996년~1999년 아주대학교 대학원 컴퓨터공학과 박사과정 수료

1991년~1996년 한국전산원 주임연구원

1997년~1998년 TTA/SC10/SG3(시스템보안 연구위원회) 간사

2000년~현재 TTA/TC10(정보보호 기술위원회) 간사, TTA/SC10/SG3(시스템보안 연구위원회) 의장

1996년~현재 한국정보보호센터 선임연구원, 표준과제 책임자

관심분야 : 컴퓨터 · 네트워크 보안, 접근통제, 정보보호 표준화

홍 기 융

e-mail : kyhong@kisa.or.kr

1985년 전남대학교 전자계산학과(학사)

1990년 중앙대학교 대학원 전자계산학과(석사)

1994년 정보처리기술사(전자계산조직응용)

1996년 아주대학교 컴퓨터공학과(박사)

1985년~1995년 한국전자통신연구소 선임연구원

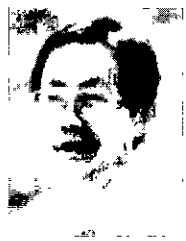
1992년~1993년 이태리, Alenia Spazio사 Senior Researcher

1995년~1996년 한국전산원 선임연구원

1996년~2000년 한국정보보호센터 응용기술팀장, 평가체계팀장, 인증관리팀장

2000년~현재 (주)케이사인 대표이사, (주)시큐브 대표이사

관심분야 : 컴퓨터 · 네트워크 보안, 정보보호 표준화, 전자상거래 보안, 전자서명인증, 공개키기반구조(PKI), Secure O. S, Linux Security



김 동 규

e-mail : dkkim@madang.ajou.ac.kr

1973년 서울대학교 공과대학 졸업(학사)

1979년 서울대학교 자연과학대학원 졸업(석사)

1984년 미국 Kansas 주립대 대학원 졸업(Ph. D, 전산학 박사, 정보통신 전공)

1981년~1982년 미국 Kansas 주립대 전산학과 교수

1979년~현재 아주대학교 컴퓨터공학과 교수

저서 : 데이터 통신시스템, 희중당, 1986년

저서 : 컴퓨터 통신 네트워크, 상조사, 1988년

한국통신학회 상임이사, 한국통신정보보호학회 부회장

관심분야 : 컴퓨터 네트워크, 정보통신 프로토콜 엔지니어링, 정보통신 Security