情 報 保 護 學 會 誌
第18卷 第5號, 2008. 10

금융 어플리케이션을 위한 효율적인 역할추출과 안전한 역할기반 접근통제 적용 방안

정 성 민*

요 약

IT기술의 변화에 따라 금융기관의 정보보호 또한 안정성을 보장하면서도 새로운 비즈니스모델에 적합한 보안대책이 요구되고 있다. 금융 어플리케이션의 보안은 정보의 기밀성, 무결성, 가용성을 만족하는 안전하고 신뢰할 수 있는 시스템과 네트워크, 그리고 보안사고에 큰 비중을 차지하고 있는 내부 사용자에 대한 적절한 권한 부여와 접근통제가 요구되어진다.

정당한 사용자가 접근하여 발생하는 보안 문제, 즉 내부자에 의한 악의적인 행위나 오용, 실수 등에 의한 기업의 피해는 외부자에 의한 의도적인 공격보다 피해 규모가 크다. 따라서 정당한 사용자로 인증을 받았다고 할지라도 업무처리에 있어서 필요한 최소한의 권한만을 부여하는 것이 필요한 것이다.

이를 위해 금융기관에 적합한 접근통제가 필요하다. 역할기반 접근통제는 적용범위가 제한적인 강제적 접근통제와 분산된 보안관리로 중앙에서 통제가 어려운 자율적 접근통제의 단점을 보완하고, 실제 업무처리에 적합한 특성을 갖는다. 하지만 기존 역할기반 접근통제를 금융기관의 다양한 금융 어플리케이션에 적용하면 다음과 같은 문제가 발생할 수 있다. 첫째, 금융 어플리케이션에서 사용되는 역할 추출 및 관리가 어렵다. 둘째, 다양한 비즈니스모델이 원하는 직무분리가 복잡하고 어렵다. 셋째, 악의적인 내부 사용자가 역할을 변조하여 과도한 권한을 가질 수 있다.

따라서 본 논문에서는 기존의 역할기반 접근통제에 인사정보 연동을 통한 효율적인 역할 추출 및 분류방안과 역할관리, 직무분리의 세분화 그리고 역할의 안전한 관리를 위해 X.509기반의 권한관리 기반구조(PMI)를 이용한 권한관리 기술을 금융 어플리케이션 환경에 효율적으로 적용하는 방안을 제시한다.

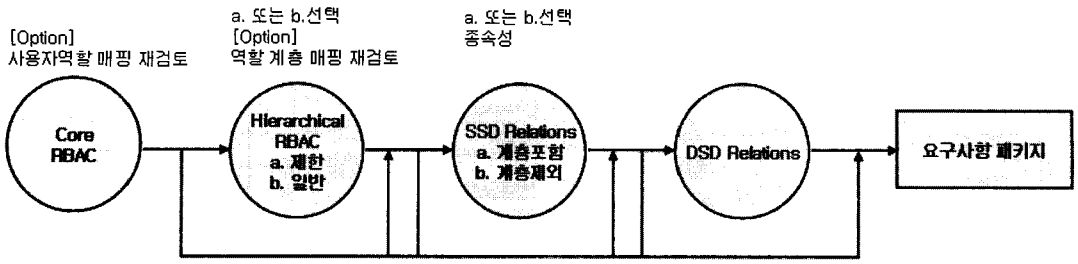
Ⅰ. 서 론

인터넷, 무선네트워크 기술의 발전과 더불어 금융환경은 인터넷뱅킹, 모바일뱅킹, 홈뱅킹 등 다양한 서비스 중심의 e비즈니스시대가 도래하고, SOX와 Basel II 등 내부통제가 강화된 비즈니스 프로세스를 요구하고 있다. 또한 금융기관의 네트워크 환경은 90년대까지 물리적으로 분리된 내부망(Intranet)을 사용하던 금융기관들이 점차 인터넷(Internet)과 내부망을 하나로 통합한 익스트라넷(Extranet)으로 바뀔 수 밖에 없었다.

이러한 IT기술의 변화에 따라 금융기관의 정보보호 또한 안정성을 보장하면서도 새로운 비즈니스모델에 적합한 보안대책이 요구되고 있다. 금융 어플리케이션의 보안은 정보의 기밀성(Confidentiality), 무결성(Integrity), 가용성(Availability)을 만족하는 안전하고 신뢰할 수 있는 시스템과 네트워크, 그리고 보안사고에 큰 비중을 차지하고 있는 내부 사용자에 대한 적절한 권한 부여와 접근통제(Access Control)가 요구된다.

완벽한 시스템 보안과 네트워크 보안을 갖추었다고 하더라도 정당한 사용자가 접근하여 발생하는 보안 문제, 즉 내부자에 의한 악의적인 행위나 오용, 실수 등에 의한 기업의 피해는 외부자에 의한 의도적인 공격보다 피해규모가 크다. 따라서 정당한 사용자로 인증을 받았다고 할지라도 업무처리에 있어서 필요한 최소한의 권한(least privilege)만을 부여하는 것이 필요하다. 이러한 사례로 최근 프랑스 소시에터 제너랄(Societe Generale)은행에서 발생한 사고는 내부 사용자 권한관리 문제에서 기인했다고 볼 수 있다. 이 은행의 트레이

* 국민은행 정보보안팀 (jsm0305@kbstar.co.kr)



〔그림 1〕 NIST RBAC 단계별 방법론

더였던 제롬 케르비엘(Jerome Kervial)은 자신의 행위를 은폐하기 위해 훔친 패스워드를 사용해 부당한 거래를 수행하였고, 이로 인해 700억 달러의 손실을 발생시켰다[2]. 이 사고는 자칫 패스워드 관리문제에 초점이 맞추어질 수 있으나 좀더 깊이 있게 살펴보면, 단순히 패스워드관리 문제가 아닌 인증된 사용자 권한관리, 특히 내부 사용자 권한관리의 중요함을 일깨워주는 사고로 볼 수 있다.

접근통제는 컴퓨터나 통신 시스템에서 비인가 된 사용, 누설, 변경, 파괴, 서비스 부인 등의 위협을 막는 것이다[7]. 접근통제는 보안등급과 규칙에 의한 강제적 접근통제(Mandatory Access Control)와 사용자가 자신의 자원을 관리하는 자율적 접근통제(Discretionary Access Control) 그리고 사용자의 역할에 따라 권한(privilege)을 부여하는 역할기반 접근통제(Role Based Access Control)로 나눌 수 있다[12].

역할기반 접근통제는 적용범위가 제한적인 강제적 접근통제와 분산된 보안관리로 중앙에서 통제가 어려운 자율적 접근통제의 단점을 보완하고, 실제 업무처리에 적합한 특성을 갖는다[12]. 하지만 기존 역할기반 접근통제를 금융기관의 다양한 금융시스템 어플리케이션에 적용하면 다음과 같은 문제가 발생할 수 있다. 첫째, 금융 어플리케이션에서 사용되는 역할 추출 및 관리가 어렵다. 둘째, 다양한 비즈니스모델이 원하는 직무분리(separation of duties)가 복잡하고 어렵다[5]. 셋째, 악의적인 내부 사용자가 역할을 변조하여 과도한 권한을 가질 수 있다.

따라서 본 논문에서는 기존의 역할기반 접근통제에 인사정보 연동을 통한 효율적인 역할 추출 및 분류방안과 역할관리, 직무분리의 세분화 및 역할의 안전한 관리를 위해 X.509기반의 권한관리 기반구조(PMI; Privilege Management Infrastructure)를 이용한 권한관리 기술을 금융시스템 어플리케이션 환경에 효율적으로 적용하는 방안을 제시한다.

Ⅱ. 관련연구

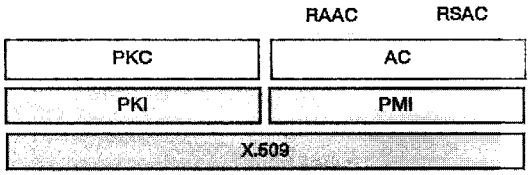
2.1 역할기반 접근통제 개요

그룹기반 접근통제에서 사용하는 그룹의 개념은 단순한 사용자들의 집합이지만, 역할기반 접근통제에서 사용하는 역할의 개념은 사용자들의 집합과 권한의 집합을 같이 고려하여 접근을 통제하는 방법이다. 역할기반 접근통제는 기존의 강제적 접근통제나 자율적 접근통제와 다르게 주체와 객체 사이에 역할이라는 개념을 두었다. 객체에 대한 사용자의 접근은 자신의 신분이 아니라 조직 내에서 가지는 직무 또는 직급 등에 따라서 결정된다. 역할기반 접근통제는 사용자의 역할에 기반을 두고 있어서 분산된 기업환경에 적합하고, 역할에 권한을 부여하는 것이 사용자 각각에 대하여 권한을 부여하는 것 보다 용이하며, 사용자 추가·삭제에 대한 운영요인이 감소한다. 하지만 한 사용자가 여러 역할을 가지고 보안에 위배되는 행위를 할 수 있는 단점이 있다. 이러한 단점은 상호배타적인 역할(Mutually Exclusive Role)을 한 사용자가 처리하지 못하도록 직무분리하여 해결하고 있다[3,8]. 실제로 Informix, Sybase, Oracle 등 많은 상용 데이터베이스 시스템과 Control-DA, DirX Identity, SAM Jupiter, Tivoli Identity Manager 등 보안관리 시스템에서 기본적인 역할기반 접근통제가 사용되고 있다[4].

2.2 NIST의 역할기반 접근통제 표준

역할기반 접근통제 기술 표준화하기 위해서 미 국가기술 표준화기구 (NIST; National Institute of Standards and Technology)에서 역할기반 접근통제에 대한 표준을 2004년에 정립하였다[1].

표준안에서는 비즈니스시스템 환경과 용도에 맞게 역할기반 접근통제 적용 단계를 구분하였다. 표준안에



〔그림 2〕 X.509 구성

제시된 내용은 다음과 같다.

Core RBAC은 사용자(User), 역할(Role), 허가(Permission), 세션(Session)으로 구성되고, 허가는 객체(Object)와 운영(Operation)의 집합이다. 추가적으로 사용자 역할매핑 재검토(User-Role Review)를 위한 요구사항을 포함한다.

Hierarchical RBAC은 역할 계층을 지원하기 위한 요구사항을 추가로 구성하고, 역할계층을 트리나 역트리 구조로 제한(Limited)하는 형태와 다중상속 개념을 포함하는 일반(General) 형태로 구분된다.

Constrainted RBAC은 Core RBAC과 Hierarchical RBAC에 직무분리를 추가하였다. 직무분리는 정적 직무분리와 동적 직무분리로 구분된다. 여기서 정적 직무분리(SSD; Static Separation of Duties)는 상호배타적인 역할을 정의하여 사용자에게 역할할당 시 제약사항을 두는 것으로 사용자는 동시에 2가지 역할을 가질 수 없고, 동적 직무분리(DSD; Dynamic Separation of Duties)는 상호 배타적인 역할을 정의하여 사용자가 역할을 사용하는 시점에 사용 제한을 두는 방식으로 사용자는 상호배타적인 역할을 가질 수 있으나 동시에 사용하지는 못한다[1,4].

2.3 권한관리 기반구조(PMI)

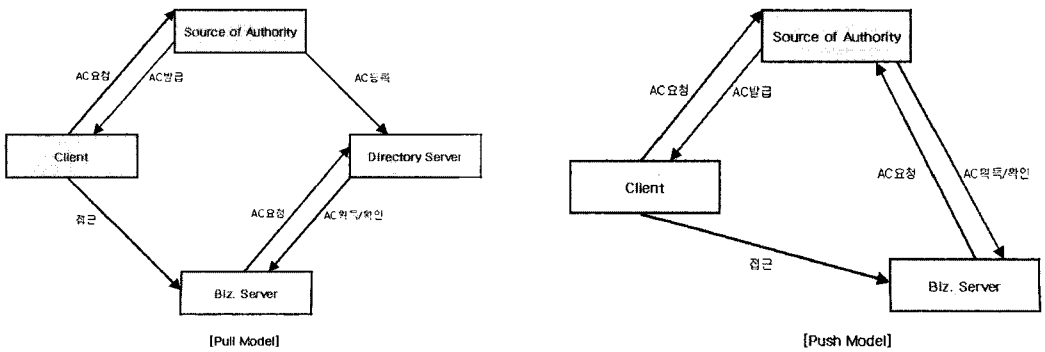
X.509 4th Edition에서 공개키 기반구조(PKI) 기술에 추가적으로 권한관리 기반구조(PMI) 기술을 제시하고 있다[6]. 공개키 기반구조(PKI)는 사용자 인증에 한정되어 있어, 인증된 사용자가 어느 시스템에 접근하여 무슨 일을 할 수 있는지에 대한 정의를 하기는 어렵다. 따라서 이러한 접근통제에 관한 정보를 권한관리 기반구조(PMI)에서 정의하는 것이다. [표 1]에 공개키 기반구조와 권한관리 기반구조의 특징을 비교하였다.

2.3.1 PMI에서 권한인증서 분배방식

권한인증서(Attribute Certificate)를 분배하는 방안은 크게 풀(Pull)방식과 푸쉬(Push)방식으로 나눈다[10].

〔표 1〕 PKI와 PMI 특징 비교

구 분	PKI	PMI
기능	- 사용자 인증관리 - 여권과 같은 의미	- 사용자 권한관리 - 비자와 같은 의미
발급	- 인증 기관에서 발급	- 권한인증관리 기관에서 발급
인증서	- 사용자 인증서PKC(Public Key Certificate)	- 권한(속성) 인증서AC(Attribute Certificate)
공통	- 공개키 암호알고리즘을 이용한 인증서 발급 및 검증	



〔그림 3〕 권한인증서 분배방식

금융기관처럼 대상 시스템이 많은 경우 디렉터리 서버에 의해 인증서를 분배하는 풀 방법이 효율적이다.

Ⅲ. 금융 어플리케이션을 위한 역할 추출 및 분류

본 장에서는 금융 어플리케이션에 역할기반 접근통제를 적용하기 위해 효율적인 역할 추출 및 분류방안을 제시한다.

3.1 역할의 기본요건 및 관리

동일한 권한을 갖는 사용자들에게 할당되는 역할은 동일하여야 한다. 그리고 한 사용자는 여러 개의 역할을 가질 수 있고, 한 개의 역할은 여러 사용자가 가질 수 있다. 또한 한 개의 역할은 여러 개의 허가(Permission)를 사용할 수 있고, 한 개의 허가는 여러 개의 역할이 사용할 수 있다.

사용자가 할당된 역할의 종류에 대한 설정은 속성할당 권한인증서(RAAC; Role Assignment Attribute Certificate)를 통해서 이루어진다. 역할이 갖는 허가의 종류에 대한 설정은 속성명세 권한인증서(RSAC; Role Specification Attribute Certificate)를 통해서 이루어지지만, 성능을 우선 시 하는 금융 어플리케이션 환경을 고려하면 DB에 역할-허가 매핑 정책테이블을 이용하여 저장 및 관리할 수도 있다.

3.2 인사정보를 이용한 역할 추출(Role Engineering)

역할을 추출하기 위해서 중요한 요소 중 하나는 인사정보 시스템과 권한관리 시스템의 관리정보 일원화다. 인사정보 시스템과 권한관리 시스템이 일원화되면 인사정보를 권한관리 정보로 이용할 수 있도록 역할을 추출할 수 있다.

금융기관뿐만 아니라 일반기업들도 어플리케이션 사용자 정보의 원천데이터는 인사정보 시스템에 근간을 둔다. 금융 어플리케이션 사용자의 입사, 퇴사, 이동, 파견, 직급, 직위, 직무 부여 및 변경 등 원천데이터를 인사정보 시스템이 최초 등록하고 관리하게 된다. 이러한 인사정보는 실제로 권한관리에 필수 정보이지만 인사정보를 그대로 금융 어플리케이션의 권한관리에 적용하는데 몇 가지 문제점이 있다. 첫째, 인사정보 시스템의 정보를 역할로 부여하는 경우 인사정보 단위로 금융 어플리케이션을 개발할 수 없다. 둘째, 모든 비즈니스를 수용할 수 있도록 인사정보를 세분화할 수 없다. 셋째, 한 사용자가 부여 받은 역할의 개수가 많아지면 역할기반 접근통제을 적용하고 관리하기 어렵다.

대규모 유럽은행(European Bank)은 종업원 수 90,000명을 두고, SAP enterprise 소프트웨어, RACF 접근통제, DB2와 LDAP(Lightweight Directory Access Protocol)을 사용하는 다양한 어플리케이션으로 구성된 IT환경에서 역할을 1,500개 추출하였고, 사용자 별로 역할 1~2개를 부여하여 운영하고 있다. 역할기반 접근통제를 사용하여 권한을 관리하는 금융기관에서 운영되고 있는 역할의 개수와 사용자 별 사용하는 역할의 개수는 [표 2]에서 확인할 수 있다[4].

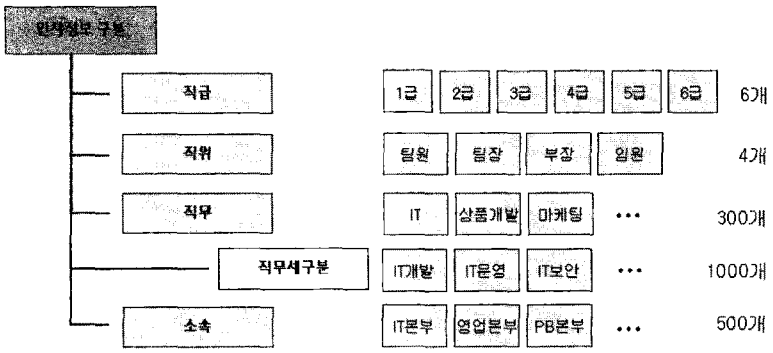
따라서 본 논문에서는 금융기관의 금융 어플리케이션 운영에 필요한 기본역할을 추출하고, 인사정보를 다양한 형태로 응용하여 역할을 도출할 수 있는 방안을 제시한다.

대부분의 금융기관은 [그림 4]과 같은 인사정보를 관리한다. 이러한 인사정보에서 추출할 수 있는 역할을 예를 들어 표시하였다. 또한 역할추출 개수를 확인할 수 있도록 각 정보 별 임의의 개수를 지정하였다.

위와 같은 인사정보를 다양하게 활용할 수 있는 방안으로 인사정보 자체를 이용하여 역할을 추출하는 기본역할과 기본역할을 응용하여 역할을 추출하는 2가지 방

〔표 2〕 기관별 역할현황

기관구분	사용자수	역할개수	사용자별 역할개수
보험사 A	167,000명	1,890개	3 ~ 5개
은행 A	31,000명	450개	약 1개
은행 B	15,000명	3,800개	약 4개
은행 C	14,500명	474개	1 ~ 2개
은행 D	90,000명	1,500개	1 ~ 2개
은행 E	186,000명	3,700개	약 5개



〔그림 4〕 인사정보 구조

식 그리고 인사정보를 기본으로 역할을 추출할 수 없는 예외적인 경우를 위한 역할 생성 및 관리방식에 따라 총 4가지로 분류하였다. 분류된 역할은 [3]에 제시된 기존의 역할기반 접근통제 정의와 [9]의 과업을 고려한 역할기반 접근통제 정의를 이용하여 분류된 역할을 재 정의하였다.

HR(C, P, J, Jd, D) ∈ Role_Engineering(R)
HR(C) ∈ R
HR(P) ∈ R
HR(J) ∈ R
HR(Jd) ∈ R
HR(D) ∈ R

3.2.1 기본역할

기본역할은 인사정보 중에서 직급, 직위, 직무, 직무세구분, 소속 등의 정보를 사용자가 사용할 역할로 추출한다. 물론 금융기관별로 관리하는 인사정보가 다를 수 있다. 비즈니스 특성과 금융기관의 환경에 맞추어 필요한 인사정보를 역할로 추출하면 된다. 인사정보에서 추출할 수 있는 기본역할이 많을수록 다음에 기술될 조합역할, 집합역할 등 복잡한 역할 추출절차를 줄일 수 있다. 기본역할은 인사정보에 해당하는 정보를 역할로 해당 사용자에게 할당하고, 인사정보 변경에 따라 자동으로 할당 및 회수한다.

[기본역할 개수]

기본역할로 도출할 수 있는 역할의 개수를 계산하면, 각각을 기본역할로 도출할 수 있으므로, 기본역할 개수 =6+4+300+1,000+500 =1,810개 의 역할이 도출된다.

[기본역할 정의]

∀ C:Class(직급), P:Position(직위), J:Job(직무), Jd:Job Detail(직무세구분), D:Department(소속), HR: HumanResource, R:Role

3.2.2 조합역할

조합역할은 기본역할로 추출된 역할을 응용하는 방법 중에 하나이다. 조합역할은 직급, 직위, 직무, 직무세구분, 소속에 포함되는 역할을 필요에 따라 2개 이상으로 묶어 AND조건으로 조합한다. 예를 들어 직위에 어떤 역할을 가지면서, 직무에 어떤 역할을 갖는 사용자를 위한 별도의 역할을 생성하는 것이다. 조합역할에서 포함된 모든 기본역할을 갖는 사용자에게만 할당하고, 인사정보 변경에 따라 자동으로 할당 및 회수한다.

[조합역할 개수]

조합역할로 도출할 수 있는 역할의 개수를 계산하면, 이항계수 ${}_nC_r(=(n!/r!(n-r)!))$에 의해 생성되는 조합의 개수를 추출하고 추출된 된 값의 합으로 각각의 역할의 개수를 계산하면,

2개씩 조합하는 경우(${}_5C_2$)는 10가지로 생성되는 역할의 개수 = 968,024개

3개씩 조합하는 경우(${}_5C_3$)는 10가지로 생성되는 역할의 개수 = 159,521,600개

4개씩 조합하는 경우(${}_5C_4$)는 5가지로 생성되는 역할의 개수 = 1,522,800,000개

5개씩 조합하는 경우(${}_5C_5$)는 1가지로 생성되는 역할

의 개수 = 3,600,000,000개,
조합역할 최대 개수 = 968,024+159,521,600
+1,522,800,000+3,600,000,000
= 5,283,289,624개
의 역할을 도출할 수 있다.

[조합역할 정의]

∀ C:Class(직급), P:Position(직위), J:Job(직무),
Jd:Job Detail(직무세구분), D:Department(소속),
HR: HumanResource, R:Role
HR(C, P, J, Jd, D) ∈ Role_Engineering(R)
HR(C) ∨ HR(P) ∈R
HR(C) ∨ HR(J) ∈R
HR(C) ∨ HR(Jd) ∈R
HR(C) ∨ HR(D) ∈R
HR(P) ∨ HR(J) ∈R
HR(P) ∨ HR(Jd) ∈R
HR(P) ∨ HR(D) ∈R
HR(J) ∨ HR(Jd) ∈R
HR(J) ∨ HR(D) ∈R
HR(Jd) ∨ HR(D) ∈R
HR(C) ∨ HR(P) ∨ HR(J) ∈ R
HR(C) ∨ HR(P) ∨ HR(Jd) ∈ R
HR(C) ∨ HR(P) ∨ HR(D) ∈ R
HR(C) ∨ HR(J) ∨ HR(Jd) ∈ R
HR(C) ∨ HR(J) ∨ HR(D) ∈ R
HR(C) ∨ HR(Jd) ∨ HR(D) ∈ R
HR(P) ∨ HR(J) ∨ HR(Jd) ∈ R
HR(P) ∨ HR(J) ∨ HR(D) ∈ R
HR(P) ∨ HR(Jd) ∨ HR(D) ∈ R
HR(J) ∨ HR(Jd) ∨ HR(D) ∈ R
HR(C) ∨ HR(P) ∨ HR(J) ∨ HR(Jd) ∈R
HR(C) ∨ HR(P) ∨ HR(J) ∨ HR(D) ∈R
HR(C) ∨ HR(P) ∨ HR(Jd) ∨ HR(D) ∈R
HR(C) ∨ HR(J) ∨ HR(Jd) ∨ HR(D) ∈R
HR(P) ∨ HR(J) ∨ HR(Jd) ∨ HR(D) ∈R
HR(C) ∨ HR(P) ∨ HR(J) ∨ HR(Jd) ∨ HR(D) ∈R

3.2.3 집합역할

집합역할은 기본역할로 추출된 역할을 응용하는 또 다른 방법이다. 집합역할은 기본역할로 추출된 인사정보에서 동일 항목에 대해서 역할을 필요에 따라 OR조건으로 추출하는 방법이다. 예를 들어 직급에서 추출한 역할을 2개 이상 묶어 OR조건으로 역할을 생성한다. 집합역할에서 제시한 기본역할 중 1개 이상을 갖는 사용자에게만 할당하고, 인사정보 변경에 따라 자동으로 할당 및 회수한다.

[집합역할 개수]

집합역할로 도출할 수 있는 역할의 개수를 계산하면, 이항계수의 총합인 2^n에서 2개 이상의 집합을 역할로 생성하므로, 각각의 집합역할의 개수는 2^n-${}_nC_1$-${}_nC_0$개의 역할이 추출된다. 따라서,
직급에서 추출되는 집합역할의 개수 = 2^6-6-1개
직위에서 추출되는 집합역할의 개수 = 2^4-4-1개
직무에서 추출되는 집합역할의 개수 = 2^{300}-300-1개
직무세구분에서 추출되는 집합역할의 개수
= 2^{1000}-1000-1개
소속에서 추출되는 집합역할의 개수
= 2^{500}-500-1개이고,
집합역할 최대 개수
= $2^6+2^4+2^{300}+2^{1000}+2^{500}$-1815개
의 역할을 도출할 수 있다.

[집합역할 정의]

∀ C:Class(직급), P:Position(직위), J:Job(직무),
Jd:Job Detail(직무세구분), D:Department(소속),
HR: HumanResource, R:Role

HR(C, P, J, Jd, D) ∈ Role_Engineering(R)
HR(C) ∧ HR(C) ∧ HR(C) ⋯ ∈ R
HR(P) ∧ HR(P) ∧ HR(P) ⋯ ∈ R
HR(J) ∧ HR(J) ∧ HR(J) ⋯ ∈ R
HR(Jd) ∧ HR(Jd) ∧ HR(Jd) ⋯ ∈ R
HR(D) ∧ HR(D) ∧ HR(D) ⋯ ∈ R

3.2.4 특수역할

금융 어플리케이션에 필요한 모든 역할을 인사정보와 이를 응용한 역할만으로 추출하는 것은 현실적으로 불가능하다. 즉, 특정 금융 어플리케이션을 개발하였는데 인사정보와는 다른 형태의 상세한 역할이 필요할 수

있다. 예를 들어 특정 금융 어플리케이션은 관리자A, 관리자B, 운영자A, 운영자B 등 인사정보보다는 상세한 역할이 필요하다. 이러한 경우를 보완하기 위해 인사정보와 무관하게 필요한 역할을 별도로 정의하여 역할을 생성한다. 이렇게 추출되는 특수역할은 인사정보와 무관하므로 별도의 역할관리를 위한 관리자를 지정하여 사용자에게 역할 부여 및 회수 등 관리 업무를 수행하여야 한다.

또한, 이런 관리부담을 최소화하기 위해서 몇 가지 제약조건(constraint)을 주어야 한다. 제약조건으로는 사용기간, IP, 또는 인사정보 변경에 따른 자동회수 등을 제약조건으로 역할을 부여할 수 있다.

[특수역할 개수]

인사정보에 의해 도출 가능한 기본역할, 조합역할, 집합역할로 정의할 수 없는 역할은 별도 생성하므로 임의의 개수가 된다.
따라서, 특수역할 개수 = α개의 역할이 도출된다.

위에서 제시한 기본역할, 조합역할, 집합역할 그리고 특수역할을 통해 금융 어플리케이션에 필요한 다양한 역할을 추출하게 되면, 추출한 역할의 총 개수는 = $5,283,287,809+2^6+2^4+2^{300}+2^{1000}+2^{500}+\alpha$개 이다.

하지만, 사용자에게 실제로 할당되는 역할의 총 개수는 = 5 + α개 이다.

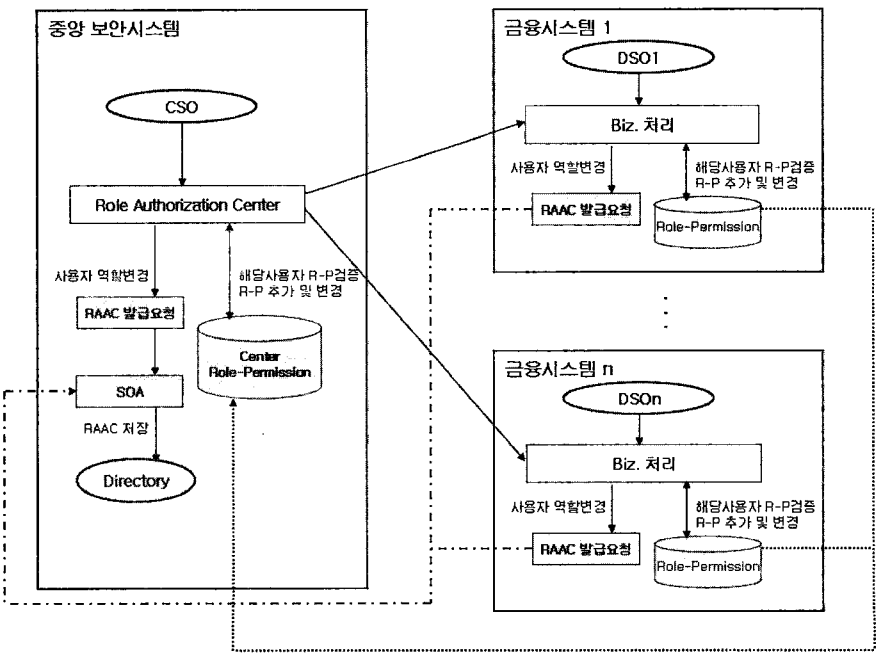
[특수역할 정의]

∀ HR: HumanResource, R:Role,
RG:Role Generating

HR(C, P, J, Jd, D) ∉ Role_Engineering(R)
RG(R) ∈ R

3.3 역할영역 구분(Role Area Division)

사용자에게 할당된 역할의 최대 개수는 5 + α이지만, 허가(Permission)를 사용할 수 있는 역할의 최대 개수는 $5,283,287,809+2^6+2^4+2^{300}+2^{1000}+2^{500}+\alpha$개 이다. 따라서 금융기관 내에서 관리하여야 하는 최대 역할의 개수는 $5,283,287,809+2^6+2^4+2^{300}+2^{1000}+2^{500}+\alpha$개 이다. 기본역할이외 조합역할, 집합역할, 특수역할은 필요에 따라 역할을 생성하여 관리하지만, 생성된 역할이 금융기관 전체 어플리케이션에서 공통적으로 사용되어지는 역할은 아니다. 따라서 어플리케이션 별로 필요한 역할만을 관리할 수 있도록 광역역할과 지역역할로 구분하는 방안을 추가로 제시한다.



[그림 5] 역할영역 구분

광역역할은 금융기관 전체 어플리케이션에서 공통으로 필요한 필수 역할을 정의하여 사용자에게 할당하고, 각 금융 어플리케이션에서 필요한 역할은 지역역할로 각자 할당 관리할 수 있도록 하고 필요 시 지역역할을 광역역할로 재정의 하도록 하여 역할 사용영역에 대한 구분을 둔다.

이렇게 함으로써 역할관리에 소요되는 관리부하를 분산하면서 광역역할의 개수를 최소화하여 궁극적으로 금융기관에서 사용하는 역할의 개수를 최소화하는 것이다.

[역할영역 구분 정의]

$\forall$ R:Role

$R \in WideAreaRole(R)$

$R \in LocalAreaRole(R)$

$WideAreaRole(R) \leq LocalAreaRole(R)$

3.3.1 광역 역할(Wide area role)

기본적으로 금융기관 내 모든 어플리케이션의 역할은 중앙 역할인가센터 (Central Role Authorization Center)가 관리한다. 사용자에게 할당되는 역할은 인사정보를 통해 추출된 역할 중 전체 금융 어플리케이션에서 공통적으로 사용되어지는 역할을 초기에 설정하고, 운영 중 발생하는 각 금융 어플리케이션의 요구를 받아 사용자 역할 할당 정보를 추가, 수정, 삭제한다.

즉, 금융기관 내 전체 금융 어플리케이션에서 필요한 역할은 중앙 보안담당자(CSO; Center Security Officer)가 중앙 역할인가센터에서 할당 및 관리한다.

3.3.2 지역 역할(Local area role)

각 금융 어플리케이션에서 사용되어지는 역할은 별도의 부서 보안담당자(DSO; Department Security Officer)가 관리한다. 부서 보안담당자는 사용자들에게 기본적으로 할당되어지는 역할 이외 필요한 역할을 정의하여 사용자에게 부여할 수 있고, 부여된 역할은 필요에 따라 역할인가센터에 광역 역할로 추가를 요청한다. 즉, 각 금융 어플리케이션에서 필요한 역할은 부서 보안담당자가 할당 및 관리한다.

Ⅳ. 금융 어플리케이션을 위한 접근통제 방향수립

본 장에서는 금융 어플리케이션에 역할기반 접근통제를 효율적이고 안전하게 적용하기 위해 단계별로 필요한 요소를 도출하여 방향수립을 위한 방안을 제시한다.

역할기반 접근통제 구축을 위한 핵심요소는 역할(Role) 추출, 허가(Permission) 도출, 사용자-역할 매핑, 역할-허가 매핑 등으로 [그림 6]과 같이 도식화 할 수 있다.

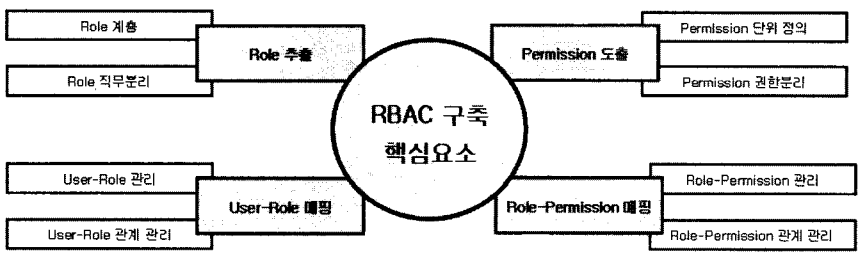
4.1 허가(Permission) 도출

허가(Permission)의 단위는 객체(Object)와 객체 사용방법(Operation)의 묶음이다. 금융 어플리케이션 개발은 비즈니스 특성에 맞게 화면단위, 메뉴단위, 버튼단위, 트랜잭션단위 등 원하는 단위로 개발하여 읽기, 쓰기, 수정, 삭제 등 사용방법을 정의하면 각각을 허가로 정의하여 개발할 수 있다.

허가는 비즈니스의 화면, 메뉴, 버튼 또는 트랜잭션 등의 단위이다. 따라서 허가는 금융 어플리케이션에 동일한 역할을 갖는 사용자가 이용할 수 있는 화면, 메뉴, 버튼 또는 트랜잭션의 최대 묶음과 동일하다.

4.2 역할과 허가(Permission) 매핑

추출된 역할과 그 역할이 사용할 수 있는 허가



〔그림 6〕 역할기반 접근통제 핵심요소

(Permission)를 매핑한다. 한 개의 역할은 여러 개의 허가에 매핑될 수 있고, 한 개의 허가는 여러 개의 역할에 매핑 될 수 있다. 즉, 역할과 허가의 매핑관계는 다:다 관계를 갖는다.

4.3 직무분리 정책

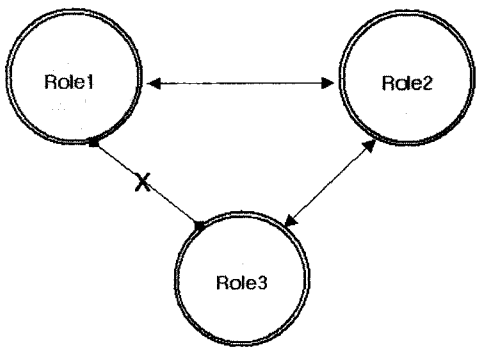
현재 개별적으로 개발 및 관리되는 금융 어플리케이션에서는 완벽한 직무분리를 기대할 수 없다. 개별 금융 어플리케이션마다 담당자가 다르고, 권한관리 체계가 다르기 때문에 담당하는 금융 어플리케이션과 타 금융 어플리케이션간 직무분리 필요성을 점검할 수 없다. 또한 역할에 대한 직무분리뿐만 아니라, 허가(Permission)에 대한 권한분리가 필요하다.

4.3.1 역할 직무분리

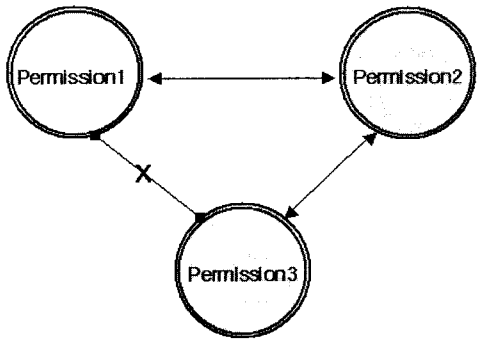
역할의 직무분리는 정적 직무분리(Static Separation of Duties)와 동적 직무분리(Dynamic Separation of Duties)로 구분하여 정의할 수 있다.

정적 직무분리는 사용자에게 역할을 할당하는 시점에 상호배타적인 역할(Mutually Exclusive Role)로 정의된 역할의 할당유무 통제를 할 수 있다. 그리고 동적 직무분리는 사용하는 시점에 사용자가 상호배타적인 역할로 정의된 여러 역할을 가지고 관련 허가(Permission)를 사용하려는 경우에 접근을 통제할 수 있다[1].

직무분리는 인사, 감사, IT, 사용자 그룹 등이 금융기관 내에서 필요한 직무분리 정책을 도출하여야 한다. [그림 7]의 경우 Role1과 Role2 또는 Role2와 Role3은 동시에 한 사용자에게 할당할 수 있으나 Role1과 Role3은 동시에 한 사용자에게 할당할 수 없다. 즉,



〔그림 7〕 역할 직무분리



〔그림 8〕 허가 권한분리

Role1과 Role3은 상호 배타적인 역할이다.

4.3.2 허가(Permission) 권한분리

역할 직무분리는 역할관점에서 상호배타적인 역할간에 사용을 금지하는 것이라면 허가(Permission) 권한분리는 역할 직무분리와 다르게 허가관점에서 상호배타적인 허가간 사용을 금지하는 방식이다. 역할기반 접근통제 관련 표준에서는 제시되어 있지 않지만, 금융 어플리케이션 환경에서는 개발자 관점에서 직무분리를 수행하는 방안이다.

허가 권한분리는 허가를 개발하고 허가를 사용할 역할과 무관하게 현재 개발된 허가와 같이 사용할 수 없는 허가를 허가 권한분리로 설정하여 허가 권한분리로 설정된 허가를 동시에 사용할 수 없도록 제한하는 방법이다. [그림 8]의 경우 Permission1과 Permission2 또는 Permission2와 Permission3은 동시에 한 사용자에게 할당할 수 있으나 Permission1과 Permission3은 동시에 한 사용자에게 할당할 수 없다.

4.4 사용자 역할할당

인사정보를 기준으로 생성되는 기본역할, 조합역할 및 집합역할은 인사정보 시스템의 갱신을 통해 자동으로 역할이 할당 및 회수되므로 권한관리를 위해 별도의 관리가 필요 없다. 하지만 특수역할의 경우 인사정보와 무관하게 운영되는 역할이므로 어플리케이션 담당자가 사용자에게 역할을 부여하거나 회수하는 등 사용자 역할할당 관리가 필요하다.

4.5 권한인증서(Attribute Certificate) 발급

4.5.1 역할할당 권한인증서(RAAC)

사용자에게 역할을 할당하게 되면 할당된 역할리스트를 안전하게 보관, 관리하기 위해 권한인증관리 기관(SOA; Source of Authority)이 사용자에게 할당된 역할리스트를 전자서명하게 된다. 이렇게 전자서명된 인증서를 역할할당 권한인증서(RAAC; Role Assignment Attribute Certificate)라 한다. 역할할당 권한인증서는 사용자 인증서(PKC)처럼 Directory Server나 DB Server 등 일반 사용자들이나 금융 어플리케이션의 접근이 용이한 공개된 저장소에 보관하게 된다.

4.5.2 역할명세 권한인증서(RSAC)

역할명세 권한인증서는 역할이 사용할 수 있는 허가(Permission) 리스트를 안전하게 보관, 관리하기 위해 권한인증관리 기관(SOA)이 역할에 할당된 허가리스트를 전자서명하게 된다. 이렇게 전자서명된 인증서를 역할명세 권한인증서(RSAC; Role Specification Attribute Certificate)라 한다. 역할명세 권한인증서는 사용자 인증서(PKC)나 역할할당 권한인증서(RAAC)처럼 Directory Server나 DB Server 등 일반 사용자들이나 금융 어플리케이션의 접근이 용이한 공개된 저장소에 보관하게 된다.

물론 금융 어플리케이션의 성능 이슈가 있는 경우에는 역할명세 권한인증서 발급 없이 안전한 DB에 역할 및 허가 매핑정보를 보관 및 관리할 수 있다.

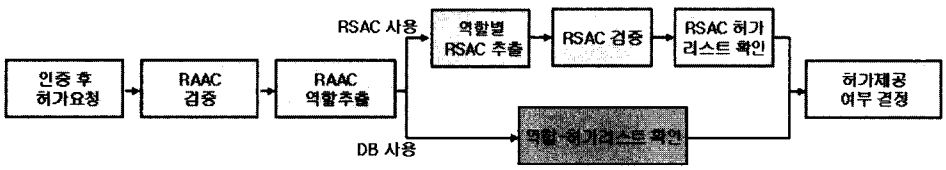
4.6 권한검증 구성

4.6.1 역할할당 권한인증서와 역할명세 권한인증서를 이용한 구성

역할할당 권한인증서와 역할명세 권한인증서를 이용하는 경우 권한검증 구성은 사용자가 허가(Permission)를 요청하게 되면, 먼저 사용자의 인증서(PKC)에 관련되는 역할할당 권한인증서를 공개된 저장소에서 추출한 다음 역할할당 권한인증서의 서명 값 및 유효기간을 검증한 후 역할리스트를 획득한다. 획득한 역할리스트의 역할에 해당하는 역할명세 권한인증서를 획득한 후 서명 값 및 유효기간을 검증한 후 역할이 사용할 수 있는 허가리스트를 확인하여 사용자에게 허가에 대한 사용 권한부여 여부를 결정하게 된다.

4.6.2 역할할당 권한인증서와 DB를 이용한 구성

역할할당 권한인증서와 DB를 이용하는 경우 권한검증 구성은 사용자가 허가(Permission)를 요청하게 되면, 먼저 사용자의 인증서(PKC)에 관련되는 역할할당 권한



〔그림 9〕 권한검증 방식에 따른 흐름

〔표 3〕 RSAC와 DB비교

구 분	RAAC-RSAC	RAAC-DB
특 징	User-Role, Role-Permission을 권한인증서로 발급하여 보안성이 높으나 성능 저하가 발생할 수 있음	User-Role은 권한 인증서로 발급하고, Role-Permission은 DB에 보관하여 성능은 향상되나 보안성이 떨어짐
User-Role 매핑	RAAC 발급	RAAC 발급
Role-Permission 매핑	RSAC 발급	DB에 매핑정보 보관
저장소	RAAC와 RSAC를 Directory Server에 보관	RAAC는 Directory Server 에 Role-Permission 매핑 리스트는 DB에 보관
보완 솔루션	HSM	DB 암호화

인증서를 공개된 저장소에서 추출한 다음 역할할당 권한인증서의 서명 값 및 유효기간을 검증한 후 역할리스트를 획득한다. 획득한 역할리스트의 역할에 해당하는 허가 사용 여부를 DB에서 역할-허가 매핑 리스트를 확인한 후 사용자가 요청한 허가에 대한 권한부여 여부를 결정하게 된다.

4.6.3 역할명세 권한인증서와 DB 사용 비교

역할명세 권한인증서와 DB를 사용하는 경우를 비교하면 [표 3]와 같다.

4.6.4 권한검증 구성방안

금융 어플리케이션은 비즈니스 성능이 중요한 요소 중 하나이다. 따라서 성능이 우수한 RAAC-DB를 이용한 권한검증체계를 구축하고, 추가적으로 DB 암호화를 통해 보안의 안정성을 보장하는 방안이 적합하다. 이 방안이 HSM(Hardware Security Module)을 통해 속도를 향상시킨 RAAC-RSAC보다 성능 및 보안성에서 우수하다.

V. 금융 어플리케이션을 위한 역할기반 접근통제 적용

5.1 역할기반 접근통제 적용을 위한 금융 어플리케이션 설계

본 장에서는 4장에서 정의된 방향을 금융 어플리케이션에 역할기반 접근통제를 실제 적용하기 위한 단계를 설명한다.

5.1.1 역할기반 접근통제 적용을 위한 업무분담

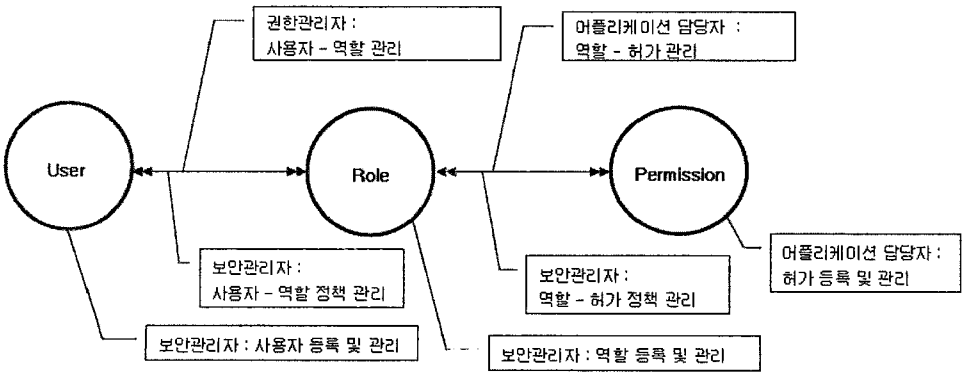
역할기반 접근통제를 적용하기 위해 금융 어플리케이션을 개발하고, 운영하는 부분과 사용자에게 역할할당 등 권한관리를 수행하는 부분 등으로 업무분담(Role and Responsibility) 정의가 필요하다. 어플리케이션 담당자는 허가를 최초 개발 시 등록 및 관리하고, 등록된 허가를 사용할 역할을 매핑하고 관리한다. 권한관리자는 실제 역할을 사용할 사용자간 매핑관계를 관리한다. 보안관리자는 역할기반 접근통제의 정책에 관련된 사항을 관리한다. 역할기반 접근통제 요소별 업무분담을 도식화하면 [그림 10]과 같다.

5.1.2 업무분담에 따른 적용플로우

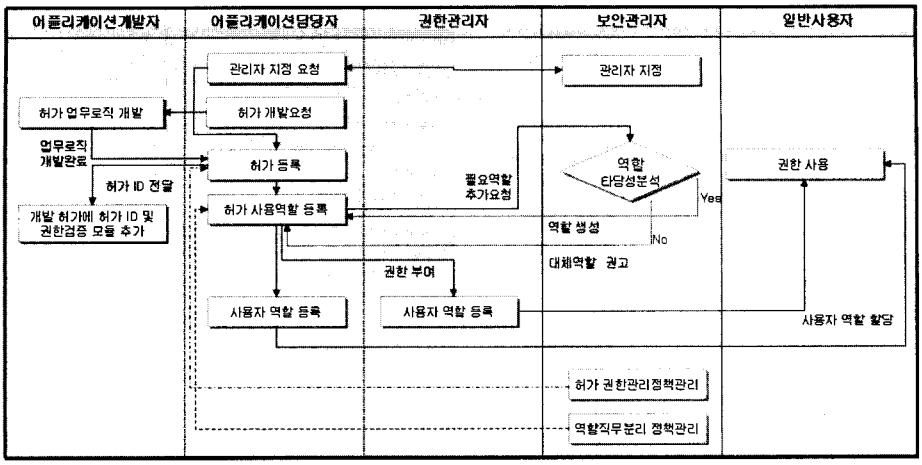
정의된 업무분담 별로 실제 역할기반 접근통제를 적용시 아래와 같은 적용플로우에 따라 허가(Permission)를 개발하고, 허가를 사용할 역할을 정의하고, 역할을 사용할 사용자를 정의하는 단계를 거치면서 금융 어플리케이션에 역할기반 접근통제를 적용한다. 업무분담별 상호연관 관계는 [그림 11]과 같다.

5.2 역할기반 접근통제가 적용된 금융 어플리케이션 접근처리 흐름도

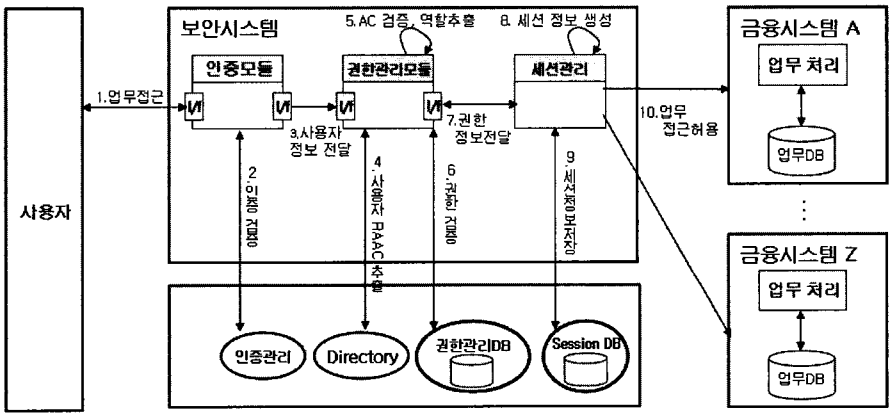
역할기반 접근통제가 적용되어 구현된 금융 어플리



〔그림 10〕 역할기반 접근통제 업무분담



〔그림 11〕 역할기반 접근통제 적용플로우



〔그림 12〕 역할기반 접근통제 처리흐름도

케이션에서 권한관리가 처리되는 과정을 [그림 12]에서 확인할 수 있다.

[처리단계별 접근시나리오]

① 사용자가 금융 어플리케이션을 사용하기 위해 접근한다.

② 접근한 사용자의 인증을 수행한다.

③ 인증된 사용자의 식별정보를 권한관리모듈에 전달한다.

④ 사용자의 역할할당 권한인증서를 Directory Server로부터 추출한다.

⑤ 역할할당 권한인증서의 전자서명을 검증하고, 사용자의 역할을 추출한다.

⑥ 사용자가 특정 허가(Permission) 요청에 대해서 권한관리모듈은 권한관리DB의 '역할-허가 매핑 리스트'를 참조하여 검증한다.

⑦ 권한이 확인된 정보를 세션 관리할 수 있도록 정보를 전달한다.

⑧ 세션관리에서 사용자의 다음 접속을 위해 세션을 생성한다.

⑨ 생성된 세션을 세션DB에 보관한다.

⑩ 사용자가 요청한 업무의 접근을 허용한다.

Ⅵ. 결 론

기존 역할기반 접근통제를 금융기관의 다양한 금융 어플리케이션에 적용하게 되면 금융 어플리케이션에서 사용되는 역할 추출 및 관리가 어렵고, 다양한 비즈니스 모델이 원하는 직무분리가 복잡하고 어렵다. 또한 악의적인 내부 사용자가 역할을 변조하여 과도한 권한을 가

질 수 있다.

이러한 문제점을 해결하고자 본 논문에서는 기존의 역할기반 접근통제에 인사정보 연동을 통한 효율적인 역할 추출 및 분류방안과 방향수립 과정에서 역할관리 방안, 직무분리의 세분화 방안 및 역할의 안전한 관리를 위해 X.509기반의 권한관리 기반구조(PMI)를 이용한 권한관리 기술을 금융기관 환경에 효율적으로 적용하는 방안을 제시하였고, 제시한 방안을 이용하여 구현한 금융 어플리케이션에서 처리 흐름도를 기술하였다.

본 논문에서는 금융시스템의 어플리케이션 관점에서 역할기반 접근통제를 적용하는 방안에 대해서 기술하였지만, 금융시스템은 어플리케이션 이외, OS 계정 및 접근관리, DB 계정 및 접근관리 등 계정(ID)이 반드시 수반되는 다양한 솔루션이 존재한다. 또한 계정이 수반되는 솔루션을 통합관리하기 위해 계정권한관리(IAM; Identity Access Management)솔루션이 출시되고 있다.

이러한 계정권한관리 솔루션에 역할기반 접근통제를 적용하는 것은 어플리케이션 관점에 역할기반 접근통제를 적용하는 것과는 다른 관점에서 적용방안에 대한 연구가 필요하다. 어플리케이션 관점에서는 사용자와 역할, 역할과 허가(Permission) 관계만 정의하여 적용할 수 있지만, 계정 관점에서는 사용자와 계정, 계정과 역할의 관계에서 대해서 추가로 정의되어야 적용이 가능하다.

따라서 향후에는 계정권한관리 솔루션에 역할기반 접근통제를 적용할 수 있는 방안과 어플리케이션 관점과 계정 관점을 통합할 수 있는 권한관리 적용방안에 대해서 연구할 예정이다.

참고문헌

[1] ANSI/INCITS 359-2004, "Information Technology-Role Based Access Control, International Committee for Information Technology Standards", 2004.

[2] Bill Brenner, "Societe Generale: A cautionary tale of insider threats", SearchSecurity.com, 2008.

[3] D. F. Ferraiolo, J. A. Cugini. D. R. Kuhn, "Role-based access control : Features and Motivations", 11th Annual Computer Security Applications Conference, 1995.

[4] Ferraiolo, Kuhn, Chandramouli, Role-Based Access Control Second Edition, pp. 326-359, pp. 171-183, pp. 213-237, ARTECH HOUSE, 2007.

[5] Gail-joon Ahn, R. S. Sandhu, "The RSL99 Language for Role-Base Separartion of Duties constraints", In Proceedings of 4thACM Workshop on Role-Based Access Control, pp. 43-54, 1999.

[6] ITU-T Rec. X.509 ISO/IEC 9594-8, "The Directory: Public-key and Attribute Certificate Frameworks", 2001.

[7] ITU-T Rec. X.812 ISO/IEC 10181-3:1996, "Security frameworks in open systems: Access control Framework", 1996.

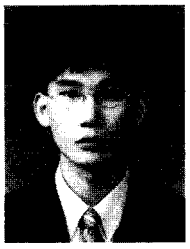
[8] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, C. E. Youman,. "Role-Based Access Control Models", IEEE Computer. Vol. 29. No. 2. pp. 38-47, 1996.

[9] Seong-Min Jeong, Seung-Wan Han, Hyeong-Seok Lim, "A Role Based Access Control Method Considering Task in the Mobile Agent-based Workflow System", ITC-CSCC(1), pp. 549-552, 2000.

[10] Venkata Bhamidipati, Ravi Sandhu, "Push Architectures for User Role Assignment", Proceedings of the 23nd National Information System, 2004.

[11] Warwick Pord, "Computer Communications Security", pp. 149-158, Prentice Hall, 1994.

〈著者紹介〉



정 성 민 (Seong-Min Jeong)

2000년 : 전남대학교 전산통계학과 석사

2000년~현재 : 국민은행 정보보안팀

<관심분야> PKI, PMI, RBAC, 분산보안