□특집□

망관리정보베이스에 대한 접근 제어 정책

김 종 덕† 정 철 윤†† 노 봉 남†††

◈ 목 차 ◈

1. 서 론

망이 개방화되고 분산화됨에 따라 여러 응용들이 지리적으로 광범위하게 분산화되어 있고 사설망이나 공중 망이 혼합된 환경에서 운용되고 있음을 감안할 때, 분산 망관리 응용들에 대하여 안전한 서비스를 제공하는 것은 필수적이다. 특히, 관리정보베이스에 저장된 정보는 망의 정상적인 운용 및 망의 유지 및 관리에 필수적인 정보들을 저장하고 있기 때문에 항상 안전하게 유지되어야 한다. 그러나 망의 보안 유지와 관련된 연구는 결함 관리나 성능 관리와 같은 다른 분야의 연구에 비하여 극히 미흡한 실정이다.

망의 정상적인 운용과 관련된 사용자들의 보안 요구 사항은 크게 인증, 접근 제어, 비밀성, 무결성, 부인 방지(non-repudiation) 등이 있다. 인증은 사용자가 합법적인 사용자인지를 검사하는 것이고, 접근 제어는 합법적인 사용자에 의해 관리 객체에 대하여 행해지는 접근을 제한하는 것을 말하고, 비밀성은 중요한 관리 정보가 불법적인 사용자에게 노출되는 것을 방지하는 것을 말한다. 무결성은 중요한 관리 정보가 불법적인 사용자에 의하여 임의로 변경되는 것을 방지하는 것을 가리키고, 부인 방지는 메시지를 보내거나 받은 것을 부인하지 못하게 하는 것을 말한다. 이러한 요구들이 전체적으로 만족되었을 때, 비로소 망의 보안이 적절하게 유지되고 있다고 할 수 있다.

본 논문에서는 망 관리에 필요한 모든 정보를 저장하고 있는 관리정보베이스에 대한 효율적인 접근 제어를 위해 3가지 접근 제어 정책을 소개하고 각각의 특성 및 각 응용분야에 대해 살펴보고자 한다.

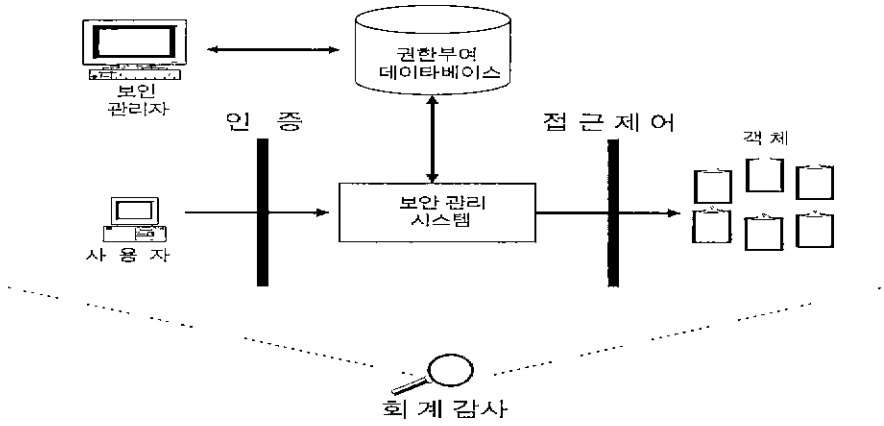
2. 접근 제어 개념

망관리 자원들을 보호하고 관리 연산들이 안전하게 수행되게 하기 위해서는 관리정보베이스에 대한 접근 제어가 확실하게 수행되어야 한다. 관리정보베이스에 대한 접근 제어의 목적은 망의 합법적인 사용자들이 수행할 수 있는 동작이나

† 정회원 : 전남대학교 전산통계학과 박사과정
†† 정회원 : 광주여자전문대학 전산정보처리학과 교수
††† 정회원 : 전남대학교 전산학과 교수

연산을 제한하므로써 망관리 정보를 안전하게 유지하는데 있다.

접근 제어는 (그림 1)처럼 여러 가지 다른 보안 서비스들과 상호 의존적인 관계에 있으며 시스템 내에 공존한다. 접근 제어는 합법적인 사용자들의 활동을 제한하는 것과 밀접한 관련이 있다. 접근 제어와 관련된 정보는 원칙적으로 보안 관리자에 의하여 관리되며, 몇몇 경우에는 사용자에 의하여 변경도 가능하다.



(그림 1) 접근 제어 및 기타 보안 서비스

(그림 1)에서 인증과 접근 제어는 분명히 구별되어야 한다. 인증은 사용자의 신원을 확인하기 위하여 필요하고, 접근 제어는 인증이 확실하게 수행되었다는 가정하에서 합법적인 사용자들에 대하여 관리 정보의 접근을 제한하기 위하여 필요하다. 따라서 접근 제어의 효율성은 사용자의 신원 확인을 위한 인증이 얼마나 잘 수행되었느냐와 각각의 사용자에게 얼마나 적절히 권한 부여가 이루어졌는가에 달려있다.

3. 접근 제어 정책

관리정보베이스의 보안 유지와 관련된 대표적인 접근 제어 정책은 크게 자율적 접근 제어(DAC: Discretionary Access Control) 정책, 강제적 접근 제어(MAC: Mandatory Access Control) 정책, 그리고 역할기반 접근 제어(RAC: Role-based Access Control) 정책 등이 있다. 이 중에서 어느 접근 제어 정책을 선택할 것인가는 관리되어야 할 환경의 특성과 그 응용에 따라 달라질 수 있다.

3.1 자율적 접근 제어 정책

자율적 접근 제어는 접근을 요청한 관리자의 신원(identification)에 근거를 두고 있다. '자율적'이라고 하는 말은 관리자가 관리 객체에 대한 접근 권한을 자율적으로 부여하거나 철회할 수 있다는 것을 의미한다. 이것은 접근 권한의 통제가 관리 객체의 각 소유자에 의하여 분산화되어 수행됨을 의미하지만, 이러한 통제는 보안 관리자에 의하여 중앙집중적으로 수행될 수도 있다.

자율적 접근 제어에서는 관리 객체에 대한 관리자의 접근 권한을 (그림 2)와 같은 접근 행렬의 형태로 표현할 수 있다.

관리 객체 / 관리자	관리 객체1	관리 객체2	관리 객체3
관리자1	M-GET M-SET M-ACTION	M-SET M-DELETE	M-GET M-ACTION
관리자2		M-GET M-SET M-DELETE	M-SET M-ACTION
관리자3	M-GET M-DELETE	M-GET M-SET M-ACTION	

(그림 2) 접근 제어 행렬

접근 제어 행렬의 행은 관리자를 나타내고 열은 관리자가 접근하고자 하는 관리 객체를 나타내며, 행과 열이 교차하는 곳은 수행 가능한 관리 연산들을 나타낸다. 예를 들면, 관리자1은 관리 객체1에 대하여 M-GET, M-SET, M-ACTION과 같은 관리 연산을 수행할 수 있다.

그러나 대규모 망에서는 수많은 관리자와 관리 객체가 존재하기 때문에 접근 제어 행렬은 그 크기가 엄청나게 커지게 되고, 반면에 접근 제어 행렬의 많은 셀들이 빈칸으로 남게 되어 매우 비효율적이다. 따라서 실제 구현 시에는 접근 제어 행렬을 이용한 방법이 거의 이용되지 않고 있으며, 대신에 접근 제어 리스트나 능력 리스트를 이용한 방법이 많이 이용된다.

먼저, 접근 제어 리스트는 접근 제어 행렬을 열을 기준으로 하여 구현하는 방법으로서, 각각의 관리 객체는 자신과 관련된 접근 제어 리스트를 하나씩 가지게 된다. 접근 제어 리스트는 각각의 관리 객체에 대하여 관리자들과 그들이 수행 가능한 관리 연산들에 대한 정보를 포함하고 있다. 접근 제어 리스트의 예가 (그림 3)에 제시되어 있다.

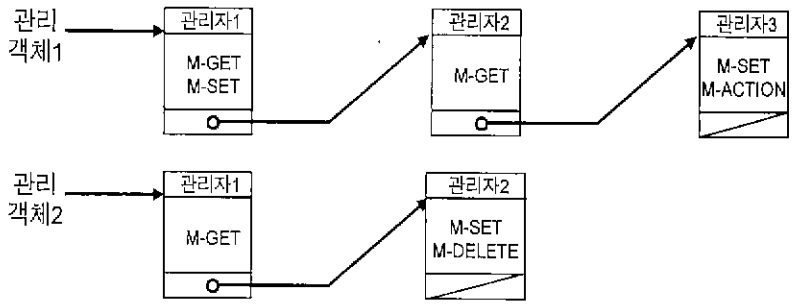
(그림 3)에서는 관리 객체1에 대하여, 관리자1은 M-GET과 M-SET 연산을 수행할 수 있고, 관리자2는 M-GET 연산을 수행할 수 있으며, 관리자3은 M-SET과 M-ACTION 연산을 수행할 수 있음을 의미한다. 또한 관리 객체2에 대해서도, 관리자1은 M-GET 연산을 수행할 수 있음을 의미하고, 관리자2는 M-SET과 M-DELETE 연산을 수행할 수 있음을 나타낸다.

한편, 능력 리스트는 접근 제어 행렬의 행을 기준으로 정렬한 것으로서, 각각의 관리자는 자신과 관련된 능력 리스트를 하나씩 가지게 된다. 즉, 능력 리스트는 각 관리자에 대하여 그들이 접근 가능한 관리 객체와 그 관리 객체에 대하여 수행 가능한 연산들의 쌍들로 이루어져 있다. 능력 리스트의 예는 (그림 4)와 같다.

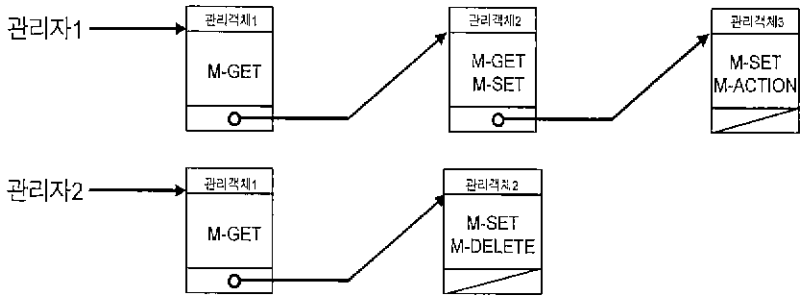
그림 4에서 관리자1은 관리 객체1에 대하여 M-GET 연산을 수행할 수 있고, 관리 객체2에 대해서는 M-GET과 M-SET 연산을 수행할 수 있고, 관리 객체3에 대하여 M-SET과 M-ACTION 연산을 수행할 수 있음을 나타낸다. 그리고 관리자2는 관리 객체1에 대해서는 M-GET, 관리 객체2에 대해서는 M-SET과 M-DELETE 연산을 수행할 수 있음을 의미한다.

접근 제어 리스트는 객체들이 세분화되어 있는 경우에 유리하며, 그리고 주체의 수가 적으며 고정적인 환경에 적합하다. 하지만 특정 관리자가 접근할 수 있는 모든 관리 객체들을 검색할 때 불편하다.

능력 리스트는 행우선에 의한 저장방식을 사용하므로써 접근 제어 리스트의 단점을 보완하였지만 객체에 대한 접근을 위해서는 모든 리스트를 살펴보아야 하는 단점은 여전히 있게 된다. 능력 리스트의 방법은 70년대에 개발되었지만 상용시스템에서 검증이 되지 않았으며 현재의 대부분의 시스템은 접근 제어 리스트의 방법을 사용하고 있다.



(그림 3) 접근 제어 리스트



(그림 4) 능력 리스트

3.2 강제적 접근 제어 정책

강제적 접근 제어는 자율적 접근 제어와는 달리 관리자와 관리 객체에 부여된 보안 등급을 기반으로 접근을 제어하는 방법이다. 각각의 관리자와 관리 객체에게는 보안 등급이 부여되며, 특히 사용자의 보안 등급을 인가 등급(clearance level)이라고도 한다.

관리 객체와 관련된 보안 등급은 관리 객체에 포함된 정보가 불법적으로 누출되었을 때 입게 되는 손해의 정도, 즉 그 정보의 중요도를 나타낸다. 그리고, 관리자와 관련된 보안 등급은 중요한 정보를 인가되지 않은 사용자에게 어느 정도로 누설하지 않을 것인가 하는 관리자의 신뢰도를 나타낸다.

자율적 접근 제어와는 달리 강제적 접근제어는 새로운 객체가 생성될 때 특정한 보안등급 부여 메카니즘에 의하여 객체에 보안등급이 부여되어야 한다. 강제적 제어정책은 모든 주체 및 객체에 대하여 일정하며 어느 하나의 주체/객체 단위로는 접근 제한을 설정할 수 없다. 즉 한 주체가 어느 한 객체를 접근하지 못하면 이때에 그 주체는 그 객체와 동일한 보안등급을 갖는 모든 객체에 접근이 허락되지 않는다.

강제적 접근 제어 정책을 이용한 대표적인 예로 BLP 모델이 있다. 정형화된 BLP모델은 보안정책을 기반으로 능동적인 주체 집합(사용자나 프로그램)에 의해 수동적인 객체 집합(화일이나 프로그

램)의 접근제어 방법이다. BLP 모델의 보안 특성은 다음과 같은 두 가지의 규칙들로 정의된다.

□ 단순 성질(simple property)
주체가 객체를 읽기 위해서는 주체의 보안등급이 객체의 보안등급을 지배해야한다. 즉 모든 (S,O,x) ∈ b에 대하여 L(S) ≥ L(O)이어야 한다.

□ *-성질(star property)
주체가 객체를 쓰기 위해서는 객체의 보안등급이 주체의 보안등급을 지배해야한다. 즉 모든 (S,O,x) ∈ b에 대하여 L(S) ≤ L(O)이어야 한다.

*-성질 특성은 높은 등급(high-level)의 데이터를 접근한 주체가 낮은 등급(low-level) 객체에 그 데이터를 기록하도록 전송할 우려가 있기 때문에 하향 기록(write-down)을 미연에 방지하기 위한 것이다. 강제적 접근 제어 정책은 정보의 비밀성이 중요한 군사 부문의 보안 정책으로는 적합하지만, 접근에 대한 통제가 너무 엄격하여 일반적인 상업 분야의 응용에는 적합하지 못한 면을 가지고 있다.

강제적 접근 제어 정책은 관리자와 관리 객체의 보안 등급을 비교하므로써 접근을 통제하는데, 다음과 같은 두 가지 기준을 기반으로 하고 있다.

- 읽기 연산 : 관리자의 보안 등급 ≥ 관리 객체의 보안 등급
- 쓰기 연산 : 관리자의 보안 등급 ≤ 관리 객체의 보안 등급

강제적 접근 제어에 의한 접근 제어는 아래와 같은 과정으로 수행되며, M-GET과 M-CANCEL-GET 연산의 경우에는 '읽기 연산'의 규칙이 적용되고, M-SET, M-ACTION, M-DELETE, 그리고 M-EVENT-REPORT 연산에 대하여는 '쓰기 연산'의 규칙이 적용된다. 그리고 M-CREATE 연산의 경우는 새로 생성된 관리 객체의 보안 등급을 관리자의 보안 등급과 같게 설정하여 준다.

```
Mac_Rule(operation, target, initiator)
{
  SELECT operation
  {
    CASE m-get, m-cancel-get   IF (LEVEL(initiator) >= LEVEL(target))
                                  RETURN TRUE;
                               ELSE
                                  RETURN FALSE;
    CASE m-set, m-action, m-delete, m-event-report :
                               IF (LEVEL(initiator) <= LEVEL(target))
                                  RETURN TRUE;
                               ELSE
                                  RETURN FALSE;
    CASE m-create : LEVEL(target) = LEVEL(initiator);
                    RETURN TRUE;
  }
}
```

강제적 접근 제어에서는 M-GET과 M-CANCEL-GET 관리 연산에 대하여는 'NO WRITE DOWN' 정책을 적용하고, M-CREATE 연산의 경우에는 새로 생성된 관리 객체의 보안 등급을 그 관리 객체를 생성한 관리 객체의 보안 등급과 같은 수준으로 할당하고 참을 반환하도록 하였다. 그리고 그 외의 관리 연산의 경우에는 'NO READ UP' 정책을 적용하여 접근을 통제하였다.

3.3 역할기반 접근 제어 정책

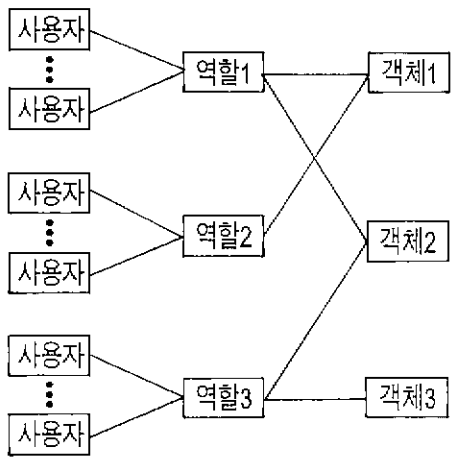
강제적 접근 제어 정책이 군대와 같은 엄격한 보안 통제를 필요로 하는 환경에서 개발되었고, 자율적 접근 제어는 학술 연구 단체와 같은 자율적이고 협동적인 환경에서 개발되었기 때문에, 두 보안 정책이 모두 상업적인 분야에 적용하기에는

다소 부적합한 면이 있다. 따라서 전통적인 자율적 접근 제어에서처럼 사용자나 사용자의 그룹에게 객체에 대한 접근 권한을 부여하고, 강제적 접근 제어에서처럼 접근 권한을 부여하는데 제한을 가할 수 있는 접근 제어 방법에 대한 연구가 진행되었다. 그 결과로서 역할기반 접근 제어 정책이 만들어지게 되었다.

역할기반 접근 제어 정책에서는 데이터 또는 객체를 몇 개의 범주로 나누었으며, 여러 명의 사용자는 역할이라고 하는 클래스들로 그룹화되어진다. 역할은 어떤 조직체의 사용자들의 임무를 여러 개의 영역으로 분할해 놓은 것으로서 역할 이름과 범주에 접근할 수 있는 권한으로 구성되어 있다. 즉, 사용자가 시스템에 대한 접근은 사용자 개인별로 권한이 주어지는 것이 아니고 공통적인 기능들에 기반을 둔 그룹들의 역할에 권한을 부여한다. 이는 사용자들을 클래스 단위로 취급하여 시스템 내에서의 사용자들을 관리하기 쉽게하는데 그 목적이 있다.

역할 기반 접근 제어는 자율적 접근 제어보다는 안전한 정보의 흐름을 보장하고, 강제적 접근 제어보다는 융통성 있는 접근제어를 제공한다. 사용자들은 그들이 수행하는 공통적인 기능들에 기반을 둔 그룹들인 역할로 구성된다. 그래서 사용자들은 시스템에 대해 접근할 때 개인별로 접근권한을 부여받는 것이 아니라 그들의 임무에 근거하여 사용자들을 그룹화한 역할에 접근 특권을 부여한다. 이렇게 하는 목적은 개개의 사용자 단위로 처리하는 대신에, 그러한 사용자들을 클래스 단위로 취급하여 개개의 사용자에 대한 권한 부여 횟수를 줄이는 것이다. 시스템 권한은 역할과 관련이 있고, 그리고 사용자도 역할과 관련이 있다. 그러므로 사용자의 접근 권한은 사용자가 어느 그룹에 속해 있는냐와 그룹이 어떠한 접근 권한을 갖고 있느냐에 달려 있다.

즉, 개개의 역할은 그 역할이 접근할 수 있는 객체 및 그 객체에 대한 접근 형태가 특권 리스트의 형태로 정의되어 있어서 실질적으로 그 역할이 수행할 수 있는 관리 범위가 능력 리스트에 의하여 제한이 된다. 각각의 관리자에게는 이렇게 정의된 역할이 적절히 배정이 되며, 이 역할에 할당된 접근 권한에 따라 관리자의 관리 객체에 대한 접근 범위가 결정된다. 역할 기반 접근 제어의 예는 (그림 5)와 같다.



(그림 5) 역할 기반 접근 제어

역할기반 접근 제어는 역할을 중심으로 접근을 통제하는데, 각각의 관리자는 자신에게 할당된 적절한 역할들을 가지고 있다. 역할에는 그 역할에 할당된 관리자 리스트와 역할이 접근 가능한 관리 객체 및 그들에 대하여 수행 가능한 연산들에 대한 정보를 포함하고 있다.

역할기반 접근 제어 정책은 자율적 접근 제어와 강제적 접근 제어의 장점을 모두 가지고 있으며, 개개의 관리자가 아닌 역할 단위로 접근을 통제하므로써 관리자의 역할 변화에 따른 접근 권

한의 감독 및 관리를 용이하게 할 수 있는 장점을 가지고 있다.

역할기반 접근 제어 정책은 아래와 같은 과정으로 수행되는데, 먼저 관리자가 속한 역할을 찾아내어 그 역할 내에 관리자가 접근을 원하는 관리 객체와 관리 연산이 정의되어 있는지를 검사하므로써 접근 허용 여부를 결정하게 된다.

```
Rac_Rule(operation, target, initiator)
{
  SELECT operation
  {
    CASE m-get, m-set, m-action, m-event-report :
                    SEARCH role CONTAIN initiator.role,
                    IF ((initiator == role.initiator IN role)
                      && (target == target IN role)
                      && (operation == operation IN role))
                      RETURN TRUE;
                    ELSE
                      RETURN FALSE;
    CASE m-delete : IF ((initiator == role.initiator IN role)
                      && (target == target IN role)
                      && (operation == operation IN role))
                    {
                      SEARCH ALL (role CONTAIN target)
                      FOR ALL role,
                        DELETE (target, operation);
                      RETURN TRUE;
                    }
                    ELSE
                      RETURN FALSE;
    CASE m-create : SEARCH role CONTAIN initiator.role;
                    ADD (target, operation) TO role;
                    RETURN TRUE;
    CASE m-cancel-get : IF (target == initiator ||
                          operation == operation IN role)
                          RETURN TRUE;
                        ELSE
                          RETURN FALSE;
  }
}
```

역할기반 접근 제어의 수행 방법은 자율적 접근 제어의 경우와 유사하지만, 먼저 접근하고자 하는 접근 주체가 자신에게 할당된 역할에 해당하는 역할기반 접근 제어 객체에 정의되어 있는가를 검사하여야 한다. 그리고 나서 그 역할 객체에 수행하고자 하는 관리 연산이 정의된 경우에만 참을 반환하고, 그렇지 않은 경우에는 거짓을 반환하여 접근을 허용하지 않는다.

4. 결 론

안전한 망관리를 위해서는 망관리에 필요한 모든 정보를 저장하고 있는 관리 정보 베이스가 여러가지 보안 위협으로부터 안전하게 유지되어야 한다. 그러나 기존의 상업용 데이터베이스 관리 시스템에서 제공하는 보안 기능은 기본적인 수준의 보안 기능만을 제공하고 있기 때문에 이러한 요구를 만족시키기에는 여러 가지 면에서 부족하다.

본 논문에서는 망 관리에 필요한 모든 정보를 저장하고 있는 관리정보베이스에 대한 효율적인 접근 제어를 위해 자율적 접근 제어 정책, 강제적 접근 제어 정책, 그리고 역할기반 접근 제어 정책 등 3가지 접근 제어 정책을 소개하고 각각의 특성 및 각 응용분야에 대해 살펴보았다.

강제적 접근 제어 정책이 군대와 같은 엄격한 보안 통제를 필요로 하는 환경에서 개발되었고, 자율적 접근 제어는 학술 연구 단체와 같은 자율적이고 협동적인 환경에서 개발되었기 때문에, 두 보안 정책이 모두 상업적인 분야에 적용하기에는 다소 부적합한 면이 있다. 따라서

전통적인 자율적 접근 제어에서처럼 사용자나 사용자의 그룹에게 객체에 대한 접근 권한을 부여하고, 강제적 접근 제어에서처럼 접근 권한을 부여하는데 제한을 가할 수 있는 접근 제어 방법으로서 역할기반 접근 제어 정책이 만들어지게 되었다.

따라서 역할 기반 접근 제어는 자율적 접근 제어보다는 안전한 정보의 흐름을 보장하고, 강제적 접근 제어보다는 융통성 있는 접근 제어를 제공한다. 이 중에서 어느 접근 제어 정책을 선택할 것인가는 관리되어야 할 환경의 특성과 그 응용에 따라 달라질 수 있을 것이다.

참고문헌

[1] Ilsoo Ahn, "Database Issues in Telecommunications Network Management," ACM SIGMOD, May 1994, pp. 37-43.

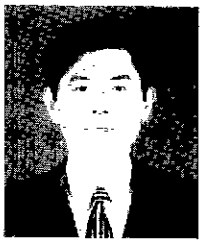
[2] David d. Clark, David R. Wilson, " A Comparison of commercial and Military computer security policies," IEEE, 1987.

[3] S. N. Bhatti, G. Knight, D. Gurle, P. Rodier, "Secure Remote Management," Proceedings of the Fourth International Symposium on Integrated Network Management, 1995, pp. 156-169.

[4] Matunda Nyanchama, Sylvia Osborn, "RoleBased Security, Object Oriented Databases & Separation of Duty," SIGMOD RECORD, Vol. 22, No. 4, December 1993, pp. 45-51.

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

[6] ISO/IEC 10164-9/ITU-T X.741, "Objects and Attributes for Access Control"



김 종 덕

1983년 전남대학교 전산학과 졸업 (이학사)
1988년 국방대학원 전자계산학과 (이학석사)
1995년-현재 전남대학교 대학원 전산통계학과 박사과정

관심분야 : 정보통신 보안, 컴퓨터네트워크, 객체지향 시스템 등

정 철 윤

1987년 전남대학교 전산학과 졸업 (이학사)
1989년 한국외대 경영정보대학원 (이학석사)
1992년 삼보컴퓨터 근무
1993년-현재 광주여자전문대학 전산정보처리학과 교수

관심분야 : 정보통신 보안, 컴퓨터 네트워크, 멀티미디어 시스템 등

노 봉 남

1978년 전남대학교 수학교육과 졸업 (이학사)
1982년 한국과학기술원 전산학과 (공학석사)
1994년 전북대학교 대학원 전산통계학과 (이학박사)

관심분야 : 정보보안, 통신망관리, 객체지향시스템, 컴퓨터와 정보사회 등