# 네트워크 자원에 대한 액세스 제어 시스템

박 형선° , 임 병렬* ,김 석우**,김 동규*

* 아주 대학교 컴퓨터 공학과
** 한국 전자 통신 연구소

# Access Control System for The Network Resource

Hyoung-Sun Park° , Byung-Ryul Lim* ,Seok Woo Kim** ,Dong-Kyoo Kim*
* Dept.of Computer Engineering , Ajou University
** Electronics and Telecommunications Research Institue

요 약

본 논문은 네트워크에서 불법적 사용자에 의한 정보의 보호를 위해, 강제적 액세스제어 정책(MAC)과 임의적 액세스 제어 정책(DAC)에 기초하여 네트워크에서의 액세스 제어 정책을 수립하고 이를 UNIX하에서 액세스 제어 리스트(ACL)와 보안 레이블에 기초하여 수행될 수 있도록 설계하고 구현하였다.

## 1. 서론

정보 통신 기술의 발전으로 각종 네트워크을 통한 공공 자원의 공유 및 필요 정보로의 액세스가 용이하게 되었다. 네트워크을 통한 각종 서비스를 제공받게 됨으로 인해 생활의 편리함을 누릴 수 있게 된 반면, 네트워크 특성상 여러 사용자에 의한 액세스가 가능함으로 해서 생기는 문제점을 내포하고 있기도 하다.

이러한 문제점은 권한이 없는 사용자가 자원에 불법적인 액세스를 시도하여 정보의 불법 변조 및 삭제, 복제 등을 행함으로 공증 정보의 신뢰성에 대한 피해를 야기케 된다.

공증 정보는 여러 사용자들을 위해 제공됨으로 신뢰성을 보장할 수 있어야 하며, 이를 위해 액세스 권한이 부여된 사용자에게 적법한 절차로 권한에 대한 승인이 이루어지도록 하여야 한다. 따라서 본 논문에서는 네트워크의 자원을 보호할 수 있도록 액세스 제어 기법에 대해 연구하였고, 강제적 액세스 제어(MAC: Mandatory Access Control ) 임의적 액세스 제어(DAC : Discretionary Access Control )정책에 기초하여, 네트워크에서의 액세스 제어 정책을 수립하였다. 그리고 이를 UNIX환경 하에서 액세스 제어 리스트(ACL)와 보안 레이블에 기초하여 수행될 수 있도록 설계하고 구현하였다

## 2. 액세스 제어 기술

### 2.1 보안 정책

#### 2.1.1 강제적인 액세스 제어(MAC : Discretionary Access Control)정책

액세스 제어는 통상적으로 강제적 액세스 제어를 의미한다. 강제적 액세스 제어는

객체에 포함된 정보의 비밀성과 이러한 비밀 성의 액세스 정보에 대하여 주체가 갖는 authorization(즉, 신원(clearance))에 근거하여 객체에 대한 액세스를 제한하는 것이다[1][2].

강제적 액세스 제어 정책은 임의적 액세스 제어 정책에 비하여 일반적으로 다음과 같은 속성을 갖는다. 첫째, 강제적 액세스 제어 정책은 객체의 소유자에 의하여 변경할 수 없는 한 주체와 한 객체간의 액세스 제어 관계를 정의한다. 둘째, 한 주체가 한 객체를 읽고 그 내용을 다른 객체에게 복사하는 경우에 원래의 객체에 내포된 강제적 액세스 제어 제약 사항이 복사된 객체에 전파(propagate)된다. 셋째, 강제적 액세스 제어 정책은 모든 주체 및 객체에 대하여 일정하며, 어느 하나의 주체/객체 단위로 액세스 제한을 설정할 수 없다. 즉, 강제적 액세스 제어가 어느 한 객체를 액세스하지 못하면, 이때에 그 주체는 그러한 특정의 비밀 등급을 갖는 모든 객체들을 액세스하는 것이 금지될 것이다.

### 2.1.2. 임의적 액세스 제어( *DAC :Discretionary Access Control*) 정책

임의적 액세스 제어는 사용자의 정보에 대한 액세스를 사용자의 ID, 사용자가 속해 있는 그룹, 객체에 대한 액세스 권한이 있는 사용자가 다른 사용자나 집단에게 그 권한을 직,간접으로 양도할 수 있는 능력 등을 기초로 제어를 가하는 것이다[1][2]. 이와 같은 임의적 액세스 제어는 주체나 또는 그들이 속해 있는 그룹들의 Identity에 근거하여, 어떠한 액세스 허가를 가지고 있는 한 주체는, 강제적 액세스 제어가 유지되지 않는 한, 임의의 다른 주체에게 자신의 허가를 넘겨줄 수 있게 된다.

임의적 액세스 제어가 갖는 일반적인 속성을 살펴보면 다음의 세 가지로 요약할 수 있다.

i) 임의적 액세스 제어 정책은 허가된 주체(즉, 객체의 소유자)에 의하여 변경 가능한 한 주체와 한 객체간의 관계를 정의한다.

ii) 한 주체가 어느 한 객체를 읽고 그 내용을 다른 어느 한 객체로 복사하는 경우에 처음의 객체에 내포된 임의적 액세스 제어 관계는 복사된 객체로 전파(propagate) 되지 않는다.

iii) 임의적 액세스 제어 정책은 모든 주체 및 객체 들간에 일정하지 않고 하나의 주체/객체 단위로 액세스 제한을 설정할 수 있다.

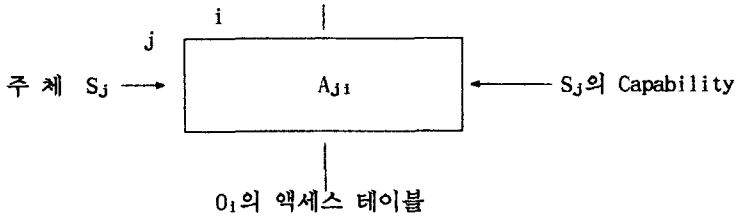
임의적 액세스 제어 정책에서 결점은 첫째, 임의적 액세스 제어 속성상 액세스는 주체의 Identity에 전적으로 근거를 두고 있으며, 둘째, 이와 같이 주체의 Identity가 매우 중요하므로 만약, 다른 사람의 Identity를 사용하여 불법 행위가 이루어질 수 있다면 임의적 액세스 제어가 파괴되어질 수 있다. 셋째, 트로이 목마에 대하여 취약하다.

## 2.2. 액세스 제어 메커니즘

### 2.2.1 액세스 행렬 (Access Matrix)

액세스 행렬은 어떤 주체와 그 주체에 의해 액세스 되는 객체, 그리고 허용되는 동작의 종류로 행렬의 구조를 표현한다. 액세스 행렬의 구조는 <그림 2.1>과 같으며 이때 주체는 행렬의 인덱스 j(행)에 의해 표현되고 객체는 행렬의 인덱스 i(열)에 의해 표현된다[2][8].

객체 $O_i$




◉ $A_{ij}$ : 주체 j가 객체 i에 대해 액세스 할 수 있는 액세스 권한.

〈 그림 2.1 〉 액세스 행렬

2.2.2 Capability 시스템

어떤 주체에 대해 액세스할 수 있는 객체와 그 객체에 대해 허용되는 동작을 표현한다. 즉, 액세스 행렬의 행을 표현한다.

▶ 장점
- 활동적인(Active) 주체들의 목록만을 주기억장치에 기억시키면 된다.
- 주체에 대해 어떤 객체들을 액세스 할 수 있는지 확인하기 쉽다.
- 특별한 객체 또는 객체 그룹에 대해서 많은 주체들이 액세스 권한을 갖는 분산 환경일 경우 Capability 시스템이 더욱 유리하다.
- Capability 시스템은 보안 서브 시스템에 좀 더 잘 적용될 수 있다.
- 보안이 부분적으로 깨질 때 그 영향을 최소화시킬 수 있다.

▶ 단점
- 주어진 객체에 대해 어떤 주체가 액세스 권한을 가지고 있는지를 확인하는 데는 비효율적이다.
- 유지가 어렵다.

2.2.3 액세스 제어 목록

Capability와는 반대로 어떤 객체에 대해 그 객체를 액세스 할 수 있는 주체와 허용되는 동작을 표현한다. 즉 액세스 행렬의 열을 표현한다.

▶ 장점

- 주어진 객체에 대해 어떤 주체가 어떤 액세스 권한을 가지고 있는지를 쉽게 확인할 수 있다.
- 어떤 주체가 다른 주체에 액세스 권한을 주거나 제거시키는 성질을 유지하기 쉽다.
- 액세스 권한의 취소가 쉽다.

▶ 단점

- 주어진 주체에 대해 그 주체가 어떤 객체를 액세스 할 수 있는지를 확인하기에는 비효율적이다.
- 매 액세스시 액세스제어 목록을 참조하여야 하므로 액세스제어 목록이 길면 이 목록을 유지, 참조하여야 하는 오버 헤드가 있다.

### 2.2.4 Capability와 액세스 제어 목록의 통합 시스템

통합 시스템은 새로운 객체에 대한 최초의 액세스는 액세스 제어 목록을 참조하고, 다음 번의 액세스를 위해 주체 자신이 임시의 Capability 목록을 유지하여 이후의 그 객체에 대한 액세스를 저장된 Capability를 이용하여 쉽게 수행한다.

## 3.액세스 제어 모델 분석

### 3.1 BLP 모델

David Bell과 Leonard Lapadular에 의해 컴퓨터 안전성 (Computer Security) 분야에서는 처음으로 개발된 수학적 모델(model)인 BLP 모델은 국가적 안전성 모델에서 가장 널리 사용되는 것으로 유한 상태 머신 모델(finite state machine model)에 근간을 둔 정규 모델(formal model)이다.

BLP 모델은 크게 세 가지의 집합(set)을 가지는데 사용자, 프로그램 (program) 혹은 프로세스(process)라고 할 수 있는 주체(subject)들의 집합과, 시스템화일(system file)이라고 할 수 있는 객체(object)들의 집합과, 액세스 모드(access mode)들의 집합 등이다[3].

각각의 주체들은 비밀인가(clearance)를 가지게 되고, 자신의 비밀인가를 넘지 않는 비밀 수준(security level)을 가질 수 있다. 각각의 객체들은 비밀 등급(security classification)을 가지게 된다.

주체가 객체에 대해 가질 수 있는 액세스 허가(access permission)는 액세스 허가 행렬로 정의하게 되는데 BLP 모델에서 기술(specify)하고 있는 액세스 모드는 다음 네 가지이다.

- ▶ *read-only* : 주체는 객체를 읽을(read) 수만 있고, 수정은 할 수 없다.
- ▶ append(write-only) : 주체는 객체에 쓸(write)수만 있고, 읽을 수는 없다.
- ▶ execute : 주체는 객체를 실행(execute) 시킬 수만 있고, 직접적으로 읽거 나 쓸 수는 없다.
- ▶ read-write : 주체는 객체를 읽고(read) 쓸(write) 수 있다.

BLP 모델의 단점으로는, 사용자가 자신의 비밀 수준이나 자신의 file에 부여된 비밀 수준을 바꾸게 되는 시스템(system)에는 부적절하며, 한 개인만으로 수행될 수 없는 운영(operation)들과 같은 요구들(requirements)을 표현하기에는 적절하지 못하다는 한계를 갖고 있다. 그리고, 디바이스(device)와 외부 인터페이스(interface)를 명백하게 언급하지 않았고, 무결성(integrity)을 제공하지 않는다는 단점을 갖고 있다.

### 3.2 래티스 모델(Lattice Model)

일반적으로 액세스 제어(access control)는 주체의(subject)의 액세스 권한(access right)을 확인하여 객체(object)의 액세스를 통제할 뿐 주체가 객체의 정보를 .사용하는 일에 대해서는 통제하지 않는다. 다시 말하면, 정보 보호(information protection)에서의 많은 문제점들이 액세스 제어 뿐만아이라 정보 흐름(information flow)에 대한 제어의 문제점 때문에 정보의 불법적인 유출이 일어나고 있다는 것이다. 래티스 모델에서는 정보 흐름 제어가 통합되어 있는 액세스 제어를 제공함으로써 정보 흐름에서의 불법적

정보 유출을 막고 있다 [6]. 정보 흐름 모델(information flow model)은 다음과 같은 다섯 가지의 구성 요소를 가지게 된다.

① 정보를 저장하는 객체들(objects)의 집합(set) (예: 파일(files),프로그램 변수(program variable), 비트(bits))
② 정보 흐름의 주체(subject)가되는 프로세스의 집합
③ 비밀 수준(security class,security level)의 집합
④ 두개의 비밀 수준으로부터 나온 정보(information)에서 binary operator에 의해 생성된 새로운 정보의 비밀 수준을 나타내는 class-combining operator
⑤ 어떠한 객체에서 다른 객체로의 정보 흐름의 허가를 결정하는 흐름 관계(flow relation)

래티스 모델의 안전성 모델을 살펴보면, 각각 주체와 객체는 비밀 수준(security level)에 대한 래티스를 가지고 있으며, 객체는 관련되어 있는 내용이나 특성으로 분류되어진 컴파트먼트(compartment)로 나누어져 있다. 또, 주체는 최소 권한 원칙(principle of least privilege)에 따라 분류된 액세스 가능한 컴파트먼트를 나타내는 래티스를 가지고 있다. 래티스 모델은 군사적 안전성 정책(military secu.ity policy)을 형식화하고 구체화한 BLP 모델을 기반으로 하고 있다.

### 3.3 액세스 행렬 모델(Access Matrix Model)

이 모델은 단순성(simplicity)과 보편성(generality), 그리고 구현 기술의 다양함 때문에 널리 사용되었다.

액세스 행렬 모델에는 3개의 주요 요소(components)가 존재한다. 수동적 객체의 집합(set of passive object), 객체를 다루는 능동적 주체의 집합(set of active subject), 주체에 의해 수행되는 객체를 관리하는 규칙 (rule)의 집합을 말한다. 모든 주체는 또한 객체라는 것은 중요한 사실이다. 그것은 다른 주체에 의해 읽혀지거나 처리(manipulate)될 수 있기 때문이다. 액세스 행렬은 주체가 열을 이루고 객체가 행을 이루는 직각 배열(rectangular array)이라고 볼 수 있다[2]. 특정 행과 열의 엔트리(entry)는 주체와 객체가 일치하는 곳의 액세스 모드(access mode)이다. 허용되는 액세스 모드는 객체의 타입(type)과 시스템의 기능성(functionality)에 의존한다. 일반적인 모드로는 read, write , append, execute가 있다.

## 4. 액세스 제어 정책 설계

액세스 제어 정책은 전체적인 안전성 정책에 따라 세워져야 하나, 전체적인 정책 부재로 인해 완전하다고 할 수는 없다. 그러나 그 나름의 안전성을 갖도록 정책을 세웠으며, 이를 전체적인 측면의 서브 셋으로 보아도 될 것이다.

### 4.1 강제적 액세스 제어 정책

[정책 1] : 읽기 액세스 정책

주체의 보안 레이블이 객체의 보안 레이블을 지배하는 경우에만 읽을 수 있다. 즉, $\lambda(S) \geq \lambda(O)$. 여기서 $\lambda$ 는 보안 등급과 범주를 나타내며, $\leq$ 와 $\geq$는 지배 관계를 나타낸다.

[정책 2] : 쓰기 액세스 정책
주체와 객체의 보안 레이블이 같은 경우에만 쓸 수 있다.
즉, λ(S) = λ(O)

### 4.2 임의적 액세스 제어 정책

임의적 액세스 제어 결정은 강제적 액세스 제어의 요구 사항을 만족한 경우에만 수행되도록 한다.

[정책 3] : 객체에 대한 ACL 부여 정책
어느 한 객체가 생성될 때, 그 객체에는 반드시 액세스 제어 리스트(ACL)가 부여된다.

[정책 4] : 액세스 모드 정책
주체는 액세스 하고자 하는 액세스 모드가 액세스모드 집합에 정의되어 있고, 객체의 ACL에 기술되어 있어야 원하는 모드의 액세스를 수행할 수 있다.

[정책 5] : ACL 권한 결정 정책
객체에 대한 ACL이 존재하지 않을 경우, MAC 정책에 따른다.

[정책 6] : ACL 변경 정책
객체의 ACL에 대한 액세스모드 권한 변경은 객체의 소유자만이 할 수 있다.

### 4.3 보안 레이블 정책

각 주체와 객체에게는 보안 등급을 식별하도록 보안 레이블이 부여되어야 한다. 이 보안 레이블은 강제적 액세스 제어 결정을 위한 기반이 된다.

[정책 7] : 보안 레이블 소유 정책
각 주체와 객체는 반드시 보안 레이블을 소유하고 있어야 한다.

[정책 8] : 보안 레이블 승계 정책
객체가 생성될 때, 객체는 주체의 보안 레이블을 승계 받는다.

[정책 9] : 보안 레이블 변경 정책
주체는 자신의 보안 레이블과 어느 한 객체의 보안 레이블을 변경할 수 없다.

### 4.4 최소 권한 정책

권한이란 제한된 서비스를 실행할 수 있는 능력을 의미한다. 최소 권한은 주체가 객체에 대하여 수행할 행위를 제한하기 위해 필요하다.

[정책 10] : 보안 레이블 변경 권한 정책
주체가 자신의 보안 레이블과 어느 객체에 대한 보안 레이블을 변경할 수

없다는 제한을 무시한다.([정책 9])

[정책 11] : 객체에 대한 ACL 부여 권한 정책
어느 한 객체가 생성될 때, 객체에 대한 ACL을 반드시 부여해야 한다는 제한을 무시한다.([정책 3])

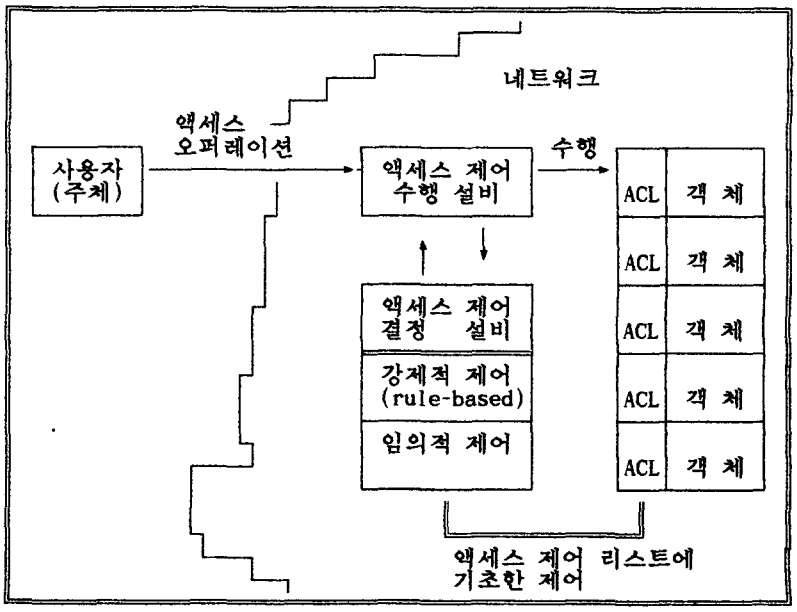
[정책 12] : ACL 변경 권한 정책
객체에 대한 ACL의 액세스 모드 변경 권한은 객체의 소유자만이 할 수 있다는 제한을 무시한다.([정책 6])

## 5. 액세스 제어 시스템 설계 및 구현

액세스 제어 시스템의 설계는 앞에서 정의한 정책들에 기반을 두었다. 여기서 쓰이는 액세스 모드는 읽기(read)와 쓰기(write)로만 한정하였으며, 오퍼레이션은 주체가 객체에 취하게 되는 행동으로 이는 다음과 같이 정의하였다.

◉ creat_object : 새로운 객체의 생성
◉ delete_object : 객체의 삭제
◉ read_object : 객체의 내용을 읽음
◉ write_object : 객체에 내용을 추가하거나, 수정
◉ change_priv : ACL의 내용을 변경

이들 오퍼레이션은 실제 환경에서 추가적으로 정의될 수 있다. 정의한 오퍼레이션이 동작하게 되는 네트워크에서의 액세스 제어 시스템의 구조는 <그림 5.1>와 같다.




< 그림 5.1 > 액세스 제어 시스템 모형

### 5.1 액세스 제어 시스템 구현

액세스 제어 시스템의 시뮬레이션 환경을 Spark/10 에서 구축하였으며, 액세스 모드는 read 와 write 두 가지만을 사용하였다. 각 주체가 객체에 대해 요구하는 오퍼레이션들은 앞장의 설계에서 정의한 대로 function으로 구현되었다. 액세스 모드와 오퍼레이션들은 실제의 시스템에 적합하도록 수정될 수도 있다. 구현된 시뮬레이션의 조건은 다음과 같다.

▶ 시스템에서 일컫는 주체에 있어 사용자라 함은 네트워크 내의 프로그래머나, 오퍼레이터를 포함하지 않는다.

▶ 각 주체는 사전에 보안 레이블을 지정 받았다. 객체 또한 보안레이블을 지정 받았으며, 새로운 객체 생성시는 그 객체를 생성한 주체의 보안 레이블을 승계 받는다. 주체는 자신 및 객체의 보안 레이블을 바꿀 수 없지만 특정 권한을 갖는(예, 관리자와 같은) 주체는 예외일 수 있다.

▶ 사용자로부터 어느 객체로의 액세스 요청시, 액세스 제어 수행 설비는 액세스 제어 결정 설비로 액세스 결정을 의뢰하고, 정당하면 액세스를 허가하게 된다.

▶ 액세스 제어 결정 설비는 우선, 규칙에 의거한 강제적 액세스 결정을 수행하고, 이차적으로 액세스 제어 리스트에 의거한 임의적 액세스 제어 결정을 한다. 만약, 어느 객체에 대한 액세스 제어 리스트가 없을 경우에는 강제적 액세스 제어 결정에 따른다.

| 오퍼레 이션 / 정책 | creat_ object | delete_ object | read_ object | write_ object | change_ priv |
|---|---|---|---|---|---|
| 1. 읽기 액세스 정책 | | | ● | | |
| 2. 쓰기 액세스 정책 | | | | ● | |
| 3. 객체에 대한 ACL 부여 정책 | ● | | | | |
| 4. 액세스 모드 정책 | | | ● | ● | |
| 5. ACL 권한 결정 정책 | | | ● | ● | |
| 6. ACL 변경 정책 | | | | | ● |
| 7. 보안 레이블 소유 정책 | ● | | | | |
| 8. 보안 레이블 승계 정책 | ● | | | | |
| 9. 보안 레이블 변경 정책 | | | | | ● |
| 10. 보안 레이블 변경 권한 정책 | | | | | ● |
| 11. 객체에 대한 ACL 부여 권한 정책 | ● | | | | |
| 12. ACL 변경 권한 정책 | | | | | ● |

< 표 5.1 > 액세스 제어 정책과 오퍼레이션간의 관계

### 5.2 액세스 제어 오퍼레이션 정의

사용자가 시스템에 액세스를 요청할 때 액세스 제어 결정은 액세스를 시도하는 오퍼레이션에 대해 수행된다. 이들 오퍼레이션에 대한 액세스 결정은 다음과 같이 정규화 될 수 있으며, 이는 시뮬레이션 구현에 적용되었다.

◉ creat_object(S,O)

$$\text{if } ( S \in SSET \text{ and } PRIV(S) \in PRIVSET )$$

$$\textit{then} \begin{cases} OSET' = OSET \cup O \text{ and } \lambda(O) \leftarrow \lambda(S) \\ \text{and } ACLSET' = ACLSET \cup ACL(O) \end{cases}$$

S,O는 각 주체와 객체이고 λ(a)는 a의 보안 레이블을, PRIV(a)는 a의 특정 권한을, ACL(a)는 a의 액세스 제어 리스트를 나타낸다. 프라임부호(')는 새로운 상태를 나타내고 <-- 는 보안 레이블의 승계를 의미하며, 각각의 SET는 각각의 집합을 나타낸다.

◉ read_object(S,O)

$$\text{if } \begin{cases} ( ( S \in SSET \text{ and } O \in OSET \text{ and } \lambda(S) \geq \lambda(O) ) \\ \text{and } ( ACL(O) \in ACLSET \text{ and } 'r' \in MOD ) ) \\ \qquad\qquad or \\ ( ( S \in SSET \text{ and } O \in OSET \text{ and } \lambda(S) \geq \lambda(O) ) \\ \text{and } ( ACL(O) \notin ACLSET ) ) \end{cases}$$

*then* can read ! ! !

◉ write_object(S,O)

$$\text{if } \begin{cases} ( ( S \in SSET \text{ and } O \in OSET \text{ and } \lambda(S) = \lambda(O) ) \\ \text{and } ( ACL(O) \in ACLSET \text{ and } 'w' \in MOD ) ) \\ \qquad\qquad or \\ ( ( S \in SSET \text{ and } O \in OSET \text{ and } \lambda(S) = \lambda(O) ) \\ \text{and } ( ACL(O) \notin ACLSET ) ) \end{cases}$$

*then* can write ! ! !

◉ delete_object(S,O)

$$\text{if } ( S \in SSET \text{ and } O \in OSET \text{ and } PRIV(S) \in PRIVSET )$$

$$\textit{then } ( OSET' = OSET - O \text{ and } ACLSET' = ACLSET - ACL(O) )$$

◉ change_priv(S,O)

$$\text{if } \begin{cases} ( S \in SSET \text{ and } O \in OSET \text{ and} \\ ACL(O) \in ACLSET \text{ and } PRIV(S) \in PRIVSET ) \end{cases}$$

$$\textit{then } ACLSET' = ( ACLSET - ACL(O) ) \cup ACL'(O)$$

## 6. 결론

네트워크라는 특성은 각종 자원에 대한 액세스가 여러 사용자로부터 쉽게 이루어질 수 있음으로 해서 생기는 문제점을 내포하고 있다.

이러한 문제점은 여러 사용자에게 신뢰성을 줄 수 있는 정보에 대한 변조 및 삭제 등을 예로 들 수 있다. 따라서,본 논문에서는 신뢰성을 기반으로 하는 정보 및 자원에 관한 액세스가 통제될 수 있도록 액세스제어 메카니즘을 연구,분석 하였으며, MAC 과 DAC 정책에 기초한 액세스 제어 정책을 수립하였다

또한, 수립된 정책을 기반으로 액세스 제어 시스템을 설계하고 이를 액세스제어 리스트와 보안 레이블을 사용해 구현하였다. 정책, 액세스모드 및 오퍼레이션을 실제 환경에 따라 추가 및 조정될 수 있도록 하였으며 , 앞으로의 연구는 정보의 흐름 제어까지 고려한 액세스제어 시스템을 설계 및 구현 하여야 할것이다.

## 참 고 문 헌


[1] "Trusted Network Interpretation", NCSC, July, 1987.

[2] "Working Draft Access Control Framework", ISO/IEC JTC1/SC21, Feb 1990.

[3] C.J. McCollum, J.R. Messing, and L. Notargiacomo, "Beyond the Pale of MAC and DAC--Defining New Forms of Access Control", Proceedings of 1990 IEEE Computer Society Symposium on Research in Security and Privacy, May 1990.

[4] Charles P. Pfleeger, "Security in Computing", Prentice-Hall, 1989.

[5] Dongkyoo Kim, Youngho Kim, Ho S. Shin, "The Design and Implementation of Common Service Element for Secure Communications in the OSI Application Layer", Proceedings of the 5th International Joint Workshop on Computer Communication, July 1990.

[6] S.T. Vinter, "Extended Discretion- ary Access Controls", Proceedings of 1988 IEEE Symposium on Security and Privacy, Apr 1988.

[7] Voydoc, V. L., Kent, S. T., "Security Mechanisms in High-level Network Protocols" , ACM 1983.

[8] 김 동규외 "정보 통신 네트워크 안전 체제의 신분 확인 및 액세스 제어 연구", 한국 전자 통신 연구소 최종 보고서, 1990.5.

[9] 김 동규외 "무선통신망 액세스제어 시스템 개발", 한국 무선국 관리사업단 최종 보고서,1994.7