PC-XINU 운영체제에서의 보안시스템의 설계 및 구현

박 용 규, 남 길 현
국 방 대 학 원

Design and Implementation of Security System on PC-XINU Operating System

Yong Kyu Park, Kil Hyun Nam
National Defense College

요 약

본 논문에서는 PC의 사용화경이 확대됨에 따른 PC보안의 필요성을 인식하고, PC에서 처리 운용되는 자료들을 보호하고 개인의 프라이버시를 보호할 수 있는 PC 보안시스템을 PC-XINU 운영체제상에서 설계하고 구현한다. 설계구현하는 보안시스템은 인증, 엑세스제어 및 정보 흐름 제어, 암호제어를 하도록하였으며, 군과같은 비밀체계에 적합하고 개인의 프라이버시를 보호할수 있도록하기위한 정책을 적용하였다. 암호알고리즘은 DES를 이용하였다.

I. 서 론

오늘날 PC(Personal Computer)는 가정을 비롯하여 산업체, 공공기관등에서 강력한 정보처리 시스템으로써 확고한 위치를 차지해 가고 있다. 또한 그 사용인구도 급격히 팽창되고 있으며, 전산망에 연결되어 그 사용폭이 확대되고 있어 이른바 정보화사회를 이룩하는데 아주 중요한 역할을 담당하고 있다고 할 수 있다. 따라서 PC내에서 처리되고 관리되는 정보를 안전하게 보호할 수 있는 필 보안성있는 PC가 요구된다고 할 수 있다.

그러나 아직까지는 PC보안이라는 용어가 생소할 정도로 PC 보안의 필요성이 크게 인식되지 못하고 있으며 특히 "개인용 컴퓨터"라는 명칭 자체가 PC가 보안과는 상관이 없는 것처럼 인식되게 하고 있다. 실제로 지금까지의 컴퓨터 보안에 대한 연구도 대형 시스템 위주로 이루어져 왔기 때문에 PC에 적합한 보안시스템도 부족한 실정이다[Stei85].

또한 PC는 물리적 접근이 용이하며 내장된 보안메카니즘의 부재, 사용자의 전문성과 책임성 부족, 감사추적기능의 부재 그리고 환경적위협이 상존하는 등의 보안적 취약성이 많기 때문에 비밀자료를 처리하거나 보관시 보안문제가 야기될 것이 분명하다.

본 논문에서는 보안성있는 PC보안 시스템을 위하여 PC-XINU 운영체제상에서 신원확인 및 인증, MAC정책과 DAC정책을 결합한 엑세스제어, DES를 이용한 암호제어를 설계 구현하고, 감사추적 기능을 제안하고자 한다.

Ⅱ. PC-XINU의 특성과 화일 시스템의 구조 고찰

PC-XINU 운영체제는 1984년 미국 퍼듀대학의 Duglas Comer에 의해서 설계된 XINU를 IBM PC에서 수행되도록 전환하여 1988년에 구현된 교육용 운영체제이다.

PC-XINU운영체제는 기본적으로 다음과 같은 특성을 가지고 있다.

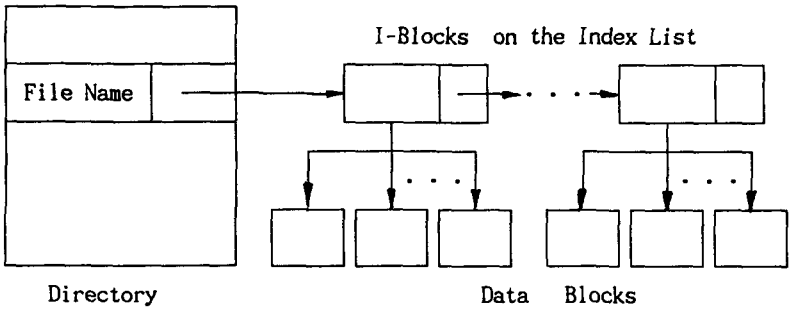
첫째, 8000라인 정도의 작은 크기의 원시코드로 구성되어 있다.

둘째, 극히 일부분을 제외하고 C언어로 작성되어 분석 및 수정이 용이하다.

셋째, 계층적구조와 기능별 모듈화로 기능의 첨가 및 수정이 용이하다.

네째, 다중프로그래밍 및 병행처리가 가능하며 분산처리를 지원할 수 있다.

PC-XINU의 화일시스템의 디렉토리 구조는 단단계구조를 갖으며 디스크블럭의 구성은 <그림 1>과 같다. 디렉토리블럭에는 화일명과 화일의 크기 그리고 해당 인덱스 블럭의 주소 등이 저장되어 있고, 인덱스 블럭에는 데이타블럭의 주소와 다음 인덱스의 주소등이 저장되고, 데이타 블럭에는 해당화일의 내용이 저장된다.



<그림 1> 디스크 블럭의 구성

화일시스템의 화일지원 프로시주어는 <그림 2>같이 다른 입출력 시스템에서와 비슷한 계층적 구조를 갖으며 각 계층별 내용은 다음과 같다.

High Level Operation은 open, close, getc, putc, read, seek, init, write, control 등 9개의 프리미티브가 있으며 사용자가 이들을 호출함으로써 화일에 대한 원하는 처리를 할 수 있다.

Pseudo-Driver Routine으로는 Pseudo-Device에 대한 입출력 프리미티브인 Upper-half Routine과 디바이스 및 콘트롤 블럭을 관리하기 위한 Control Routine이 있으며, 인덱스를 관리하기 위한 Index Manager프러시주어로는 ibclear, ibget, ibnew, iblfree, ibpit 등이 있다.

Disk Device Driver Routine으로는 Disk Device의 입출력 프리미티브인 Upper-half Routine, 디스크 스케쥴링을 위한 Optimize Routine, Interrupt Handler를 호출하는 Lower-Half Routine으로 dsinter가 있다.

Interrupt Handler는 판독을 위한 dread, 기록을 위한 dwrite, 논리주소를 물리주소로 전

환하는 dio, BIOS호출로 수행되는 dskio 프로시주어가 있다.

<table>
<tr><td colspan="2">User process</td></tr>
<tr><td colspan="2">High-Level Operation</td></tr>
<tr><td rowspan="3">Pseudo-Device driver
Routine</td><td>Upper-half Routine</td></tr>
<tr><td>Conntrol Routine</td></tr>
<tr><td>Index Manager</td></tr>
<tr><td rowspan="3">Disk Device Driver
Routine</td><td>Upper-half Routine</td></tr>
<tr><td>Optimize Routine</td></tr>
<tr><td>Lower-half Routine</td></tr>
<tr><td colspan="2">Interrupt Handler</td></tr>
<tr><td colspan="2">Hard disk</td></tr>
</table>

<그림 2> PC-XINU 화일 시스템의 계층구조

Ⅲ. 엑세스제어 정책

엑세스제어는 주체인 사용자 또는 프로세스가 객체인 화일에 대하여 접근하려 할 때 객체의 사용권이 있는 정당한 주체인가를 확인하여 엑세스 허용여부를 결정함으로써 비인가자의 불법적 엑세스를 통제하기 위한 것이다. 따라서 적절한 엑세스 제어는 비밀성, 신뢰성에 대한 위협으로부터 자료를 보호할 수 있다[Denn87]. 엑세스제어를 위한 정책으로는 MAC(Mandatory Access Control)정책과 DAC(Discretionary Access Control)으로 구분해 볼 수 있다. TCSEC(Trusted Computer System Evaluation Criteria)에서는 C분류의 엑세스제어정책으로 DAC정책을 만족할 것을 요구하고 있으며, B분류를 위해서는 MAC정책을 수용하도록 요구하고 있다. 본 장에서는 MAC정책과 DAC정책을 간단히 설명하고자 한다.

1.MAC 정책

TCSEC에서 MAC정책은 다음과 같이 정의된다.

"객체에 포함된 정보의 기밀성(sensitivity)과 이러한 기밀성의 엑세스정보에 대하여 주체가 갖는 formal authorization(즉 신원(clearance))에 근거하여 객체에 대한 엑세스를 제한하는 방법을 MAC라고 한다"[Mead83].

MAC 정책에서 엑세스를 위한 주체 객체 사이에 존재해야 할 조건은 다음과 같다.

- 주체가 객체에 대한 read엑세스 조건

  주체의 비밀수준이 객체의 비밀수준보다 크거나 같고, 주체의 비 계층적 카테고리들이 객체의 비 계층적 카테고리를 포함하는 경우.

- 주체가 객체에 대한 write엑세스 조건

주체의 비밀수준이 객체의 비밀수준보다 작거나 같고, 주체의 비 계층적 카테고리들이 객체의 비계층적 카테고리를 포함하는 경우.

이러한 MAC정책은 객체의 소유자에 의하여 변경할 수 없는 주체와 객체간의 엑세스제어관계를 정의하며, 모든 주체와 객체에 대하여 일정하게 적용되고, 어느 한 주체나 객체단위로 엑세스제어 제한을 설정할 수 없다.

2. DAC 정책

TCSEC에서 DAC정책은 다음과 같이 정의된다.

"주체나 또는 그들이 속해있는 그룹들의 식별자(ID)에 근거하여 객체에 대한 엑세스를 제한하는 방법을 DAC라고 한다. 이때 엑세스제어는 객체의 소유자의 임의적(discretionary)인 판단에 의하여 이루어진다."

DAC정책은 허가된 주체(객체의 소유자)에 의하여 변경가능한 하나의 주체와 하나의 객체간의 관계를 정의하고 모든 주체 및 객체들 간에 일정하지 않으며 주체/객체단위로 엑세스 제한을 설정할 수 있다. DAC의 결점은 DAC의 속성상 엑세스는 주체의 ID에 전적으로 근거를 두고 있으며 데이타의 의미나 속성에 근거하여 엑세스를 결정하지 못한다는 점이다.

Ⅳ. PC보안 시스템의 설계 및 구현

1. 설계시 고려사항

PC보안 시스템은 개인이 사용하는 PC내의 자료들에 대한 불법적 접근 위험을 방지하고, 사무실내에서 공동으로 사용될 경우 공동 및 개인 화일의 불법 사용 및 누출로부터 보호하기 위한 시스템이다. 따라서 PC보안시스템은 시스템자체의 적절한 물리적 보호 대책이 전제되어야 하며 요구되는 보안수준을 제공하여야 하고 사용자의 편의와 사용의 효율성, 그리고 PC의 특성상 여러 환경에서 적절히 사용될 수 있는 범용성이 고려되어야 한다.

2. 구현 환경 및 범위

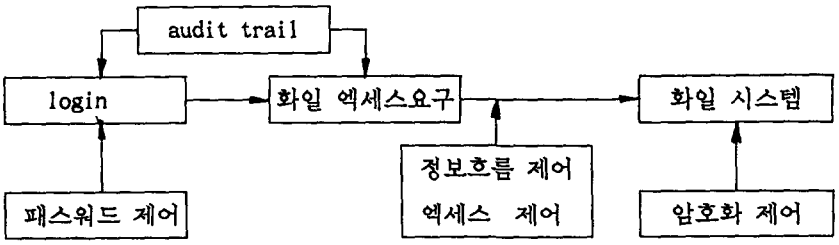
PC보안의 메카니즘 구현은 응용소프트웨어로 구현하거나 운영체제내에서 운영체제의 일부로 구현할 수 있다. 응용소프트웨어로 구현하여 적용할 경우에는 사용자가 프로그램을 통하여 직접 시스템 루틴을 호출함으로써 보안 메카니즘을 우회하는 것이 가능할 수가 있다.

본 논문에서는 보안 메카니즘을 운영체제상에서 하나의 커널화하여 모든 엑세스를 감시하여 제어하는 참조 모니터(reference monitor)의 형태로 PC-XINU운영체제상에서 구현하고, 터보 C언어를 사용하고자 한다.

본 논문에서는 신원확인 및 인증 메카니즘, 엑세스제어 메카니즘, 암호화를 중점 구현하며, 감사추적 기능은 구체적 설계와 구현을 생략하고자 한다.

3. PC보안 시스템의 메카니즘 설계 및 구현

본 논문에서 설계하고자 하는 PC보안 시스템은 여러가지 보안기법을 혼합한 다단계 메카니즘을 사용한다. 즉 <그림 4>와 같이 사용자가 로그인시 신원확인과 인증을 실시하고, 엑세스권한을 체크하여 불법적인 화일의 사용을 통제하며, 동시에 정보흐름을 제어한다. 또한 화일을 암호화 상태로 저장하여 보호할 수 있게 하며, 감사추적(audit trail)기능을 적용하는 다단계 제어모델을 설정하였다.



<그림 4> PC보안 메카니즘의 개념도

가. 신원확인(Identification) 및 인증(Authentication)

사용자의 신원확인과 인증은 로그인(log-in)시 실시되어 적법한 사용자인가를 확인하여야 한다. 본 논문에서는 3가지의 정보 즉 비밀이 아닌 식별자 ID와 비밀 암호인 패스워드(password), 사용자 신원허가(clearance)를 사용하였다. 신원허가는 사용자가 로그인 할 수 있는 최고의 등급 UML(User Maximum Level)로 정의 한다. 인증정보의 안전한 보호를 위해서는 개인정보를 <그림 5>와 같이 개인정보테이블에 기록하여 <그림 6>과 같은 자료구조를 갖는 디스크의 시스템 블럭에 저장한다. 이때 비밀로 유지해야 하는 패스워드 및 개인고유키(private key)는 단방향함수로 암호화하여 저장한다.

ID	f(pwd)	UML	private key	user category	check No
A	hjjl;jlk	2	jjklkkll	a,c	10
B	bijkjerm	1	hkkrtyrf	a,b,d	8
C	potrhmng	3	dvbbui,r	b,d	21

<그림 5> 개인정보 테이블

```
struct  sysblk  {                       /* system block layout      */
        char    user_id[10][5];         /* user_id ( 5 * 10 )       */
        char    pass_id[10][8];         /* password ( 8 * 10 )      */
        int     grade[10];              /* securite grade           */
        char    pskey[10][8];           /* private key              */
        int     group[10];              /* group                    */
        int     checknum[10];           /* password check number    */
        char     mastk[8];              /* master key               */
}
```

<그림 6> 시스템블럭의 자료구조

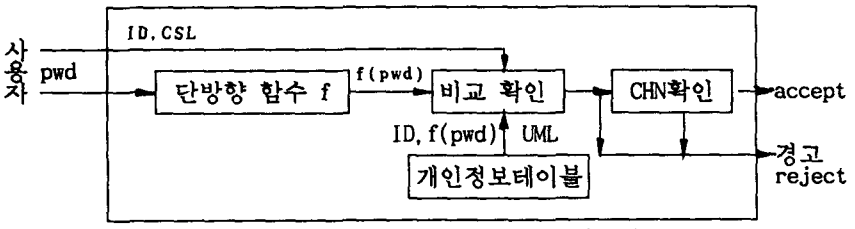
신원확인 및 인증의 메카니즘은 <그림 7>과 같으며, 그 절차는 다음과 같다.

1) 사용자 ID 제출

2) 사용자의 Password 제출

3) 사용자가 접근 하고자 하는 비밀 수준(Current Security Level:CSL) 제출

4) 개인정보테이블에 ID가 존재하고 패스워드가 일치하며 UML>=CSL 이면 accept.



ID : 식별자, pwd : 패스워드
CSL: Current Security Level, CHN: 체크번호
UML: User Maximum Level

<그림 7> 인증 메카니즘

인증을 위한 패스워드의 비교시에 패스워드의 반복 사용에 의한 노출위협을 줄이기 위해, 각 사용자의 로그인 횟수를 개인정보테이블에 유지하여 일정횟수 이상 사용시 경고하고 변경을 지시하도록 하였으며, 한계횟수에 도달시 더이상 기존의 패스워드로는 로그인을 할 수 없도록 하였다.

패스워드의 등록 및 관리절차는 패스워드의 보안에 큰 영향을 미치므로 본 논문에서는 개인정보테이블의 관리는 오직 SA(system administrator)만이 접근할 수 있는 테이블관리모듈인 tmm()을 구현하였다. SA가 테이블 관리모듈을 통하여 등록한 후 사용자에게 제공하면, 사용자는 패스워드를 자신만이 아는 것으로 변경하여 사용하도록 chpwd()루틴을 구현하였다.

재인증은 relog() 명령루틴을 구현하여 사용자가 컴퓨터 사용후에 relog명령을 입력하면 새로운 사용자의 인증준비상태가 되도록 하였다. 인증시 사용되는 사용자 ID와 로그인 수준(CSL)은 전역변수에 할당되어 사용자의 재인증 명령전까지 제어요소로 사용된다.

나. 화일 엑세스제어

본 논문에서는 군과 같은 특수한 비밀 관리체계를 갖는 조직의 필요에 부합된 엑세스제어와 정보의 흐름을 제어할 수 있는 MAC정책과, 소유자의 재량권에 의해 엑세스를 제어 함으로써 개인의 프라이버시를 보장할 수 있는 DAC정책을 결합한 형태의 모델을 설정하였다.

(1) 용어의 정의

모델을 정의하기 위한 용어를 정의하면 다음과 같다.

- 비밀 수준(Security Level(SL)) : 보안성 또는 보안요구 정도로써 1급, 2급, 3급, 4급, 5급의 5개 등급으로 분류 한다.

- 객체의 비밀수준(OSL): 객체에 부여된 SL을 말한다.
- 주체의 비밀수준(SSL) : 주체에게 부여된 비밀취급인가(clearence)를 말한다.
- CSL(Current Security Level): 주체가 login시 제시하는 등급이다.
- 카테고리(Category(C)) : 주체 및 객체가 관련있는 업무분야로써 A 에서 E까지 5개의 카테고리를 정의하며 주체와 객체는 복수의 카테고리에 속할 수 있다.
- 카테고리집합(Category Set:SCS,OCS): 주체 및 객체가 속한 카테고리들의 집합이다.
- 비밀등급 : SL과 CS을 원소로 하는 Set이다. 비밀등급 A = {SL,CS}
- 보안 레이블(Security Label): 주체 및 객체에게 할당된 비밀등급표시를 말한다.

(2) 변형된 MAC정책의 설계

기존의 MAC정책에서는 주체의 카테고리가 객체의 카테고리를 포함하는 경우에만 엑세스가 허용되므로 결국 수직적인 카테고리 또는 광범위한 지역적 카테고리에 적합하다. 그러나 PC는 사무실 또는 LAN을 통한 사무실간의 공유가 이루어지므로 MAC정책을 PC사용 환경에 적합하도록 변경 되어져야 한다. 이를 위해 본 논문에서는 MAC정책의 엑세스 조건을 다음과 같이 변경하였다.

- read 엑세스

주체의 현재 비밀수준이 객체의 비밀수준보다 크거나 같고 주체의 카테고리와 객체의 카테고리가 상호 공통 부분이 있을 경우 즉 CSL>=OSL 이고 $SCS \cap OCS \neq \phi$ 일 경우만 허락된다.

- write 엑세스

주체의 현재 비밀수준이 객체의 비밀수준보다 작거나 같고 주체의 카테고리와 객체의 카테고리가 상호 공통부분이 있을 경우 즉 CSL<=OSL 이고 $SCS \cap OCS \neq \phi$ 일 경우에만 허락된다.

(3) DAC정책의 설계

본 논문에서 DAC정책은 화일 엑세스 모드를 'O'(소유자)와 'G'(그룹)으로 구분하여 소유자에게만 엑세스하게 하거나, 또는 화일과 연관된 카테고리를 갖는 자에게만 엑세스하게하는 제어 정책을 갖도록 하였다. 화일 생성자(소유권자)는 화일 생성시 엑세스모드와 화일을 엑세스할 수 있는 카테고리와 read 또는 write 카테고리를 지정하게 함으로써 소유권에 의한 엑세스제어가 달성되도록 하였다.

따라서 엑세스 모드가 'O'인 경우는 엑세스하고자 하는 자가 화일의 소유자이면 read, wriite 엑세스가 가능하게 하며, 'G'인 경우의 엑세스 조건은 다음과 같다.

- read 엑세스

주체의 카테고리와 객체의 read권한 카테고리가 상호 공통부분이 존재하는 경우에만 허락한다.

- write엑세스

주체의 카테고리와 객체의 write권한 카테고리가 상호 공통부분이 존재 하는 경우에만 허락한다.

(4) MAC와 DAC의 결합

두 정책의 결합은 MAC와 DAC를 동시에 만족할 때 엑세스를 허용하게 함으로써 이루어진다. 즉 MAC정책에 통과한 자라도 DAC정책에 위배되지 않아야 엑세스를 할 수 있다. 특히 화일의 카테고리를 화일의 소유자가 부여할 수 있기 때문에 소유자의 권한에 의한 엑세스 권한부여 및 배제가 가능해진다.

(5) 보안 레이블의 설계

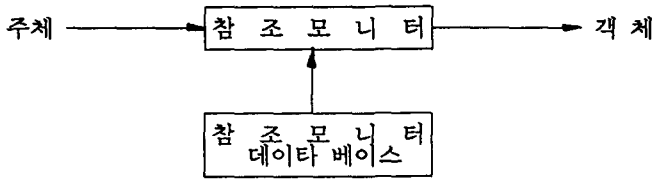
엑세스제어 정책을 실현하기 위해서는 먼저 주체와 객체의 보안레이블을 안전하게 유지할 수 있어야 한다. 이러한 주체와 객체의 레이블은 보안커널이 엑세스허가를 결정하기 위한 정보로 사용된다.

주체의 보안 레이블은 앞에서 설명된 바와 같이 개인정보 테이블에 기록하여 디스크의 시스템블럭에 저장하도록 하고, 테이블을 안전하게 보호되기 위해서 SA에게만 관리할 수 있도록 하였다.

또한 각 객체는 객체가 담고 있는 정보의 기밀성에 따라 이에 적합한 비밀수준과 카테고리가 부여되어야 하는데 본 논문에서는 객체 보안레이블을 위해 화일헤더를 확장하여 각 화일별로 저장 하도록 하였다.

(6) 엑세스제어 메카니즘의 설계 및 구현

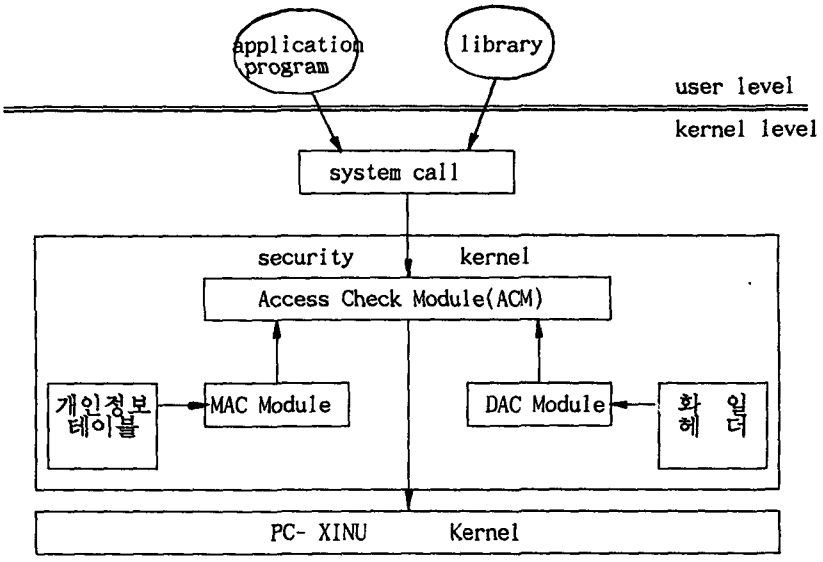
엑세스제어 메카니즘은 참조 모니터(reference monitor) 개념을 적용하여 설계하였다. 참조 모니터는 <그림 8>과 같이 주체의 객체에 대한 모든 엑세스를 모니터하여 엑세스허가를 결정하는 개념적 모델이다.



<그림 8> 참조모니터의 개념

이러한 참조 모니터개념을 기반으로한 엑세스제어 메카니즘을 <그림 9>와 같이 설계하였다. 주체가 응용프로그램(application program)이나 라이브러리 함수(library function)를 통해서 화일 접근을 위한 시스템 함수(system function)들을 호출(call)할 때 반드시 엑세스 체크모듈(ACM:Access Check Module)을 거치도록 하여 주체의 객체에 대한 엑세스 권한을 확인하도록 하는 하나의 보안커널로 구성하였다.

보안커널은 주체가 요구한 엑세스에 대해 엑세스 허가여부를 결정할 수 있도록, 엑세스정책에 부합되는지 확인하는 MAC모듈(MACM:Mandatory Access Check Module) 및 DAC모듈(DACM: Discretion-ary Access Check Module)과, MAC와 DAC의 결과를 가지고 엑세스허가를 결정하는 ACM으로 구성된다.



<그림 9> 엑세스제어를 위한 보안커널 메카니즘

- MAC 모듈

MAC 모듈은 <그림 10>의 알고리즘과 같이 앞에서 정의한 엑세스 조건에 부합 되도록 주체와 객체의 비밀 수준을 비교하고, 카테고리를 비교하여 공통부분이 있는가를 확인 하도록하였다. 이때 카테고리의 비교연산을 쉽게하기 위하여 parsegroup()루틴을 구현하여 이용하였다. parsegroup()은 a,b,c,d,e를 각각 8진수(octal)인 01, 02, 04, 010, 020를 할당하며, 복수의 카테고리일때는 각각을 비트별 OR연산하여 하나의 비트열을 생산하는 루틴이다. 따라서 카테고리간의 공통부분 존재여부는 비트별 AND연산으로 쉽게 확인된다.

```
MAC()
{ if(mode = r) and (CSL>=OSL) and ((SCS & OCS) !=0))  return(true);
  if(mode = w) and (CSL<=OSL) and ((SCS & OCS) !=0))  return(true);
  if(mode = rw) and (CSL=OSL) and ((SCS & OCS) !=0))  return(true);
    else return(false);
}
```

<그림 10> MAC 모듈의 알고리즘

- DAC모듈

DAC모듈을 구현하기 위하여 화일 헤더에 accessmod를 두어 'O'와 'G'로 화일생성자가 나타낼 수 있도록 하였다. accessmod 가 'G'인 경우를 위하여 화일 헤더의 struct group 자료구조에 rbit와 wbit를 두어 화일 생성자가 read권한이 있는 카테고리들과 write권한이 있는 카테고리들을 나타낼 수 있도록 하였으며, 이를 참조하여 DAC엑세스 조건을 확인하도록 하였다. DAC모듈의 알고리즘은 <그림11>과 같다.

```
DAC()
 { if((accessmod='O') and (curusr=author))  then return(true);
   if (accessmod='G')
     if((mode='r') and (rbit & group(curusr))!=0)) then
               return(true);
     if((mode='w') and (wbit & group(cursur))!=0)) then
               return(true);
     if((mode='rw') and (rbit & wbit & group(cursur))!=0)) then
               return(true);
  else return(false);
```

<그림 11> DAC모듈의 알고리즘

- 엑세스 체크모듈(ACM)

ACM에서는 MAC과 DAC을 이용하여 주체의 엑세스요구에 대한 허락 여부를 결정한다.

ACM의 처리 알고리즘은 다음과 같다.

ACM() = true if(MAC() = true) and (DAC() = true)

false otherwise

따라서 MAC()과 DAC()이 모두 true일때 엑세스권한을 획득하게 된다.

ACM의 수행환경은 dfdsrch루틴이 된다. dfdsrch는 디스크화일을 open시키는 dsopen의 요청에 의해 디렉토리 블럭에서 해당 화일을 찾는 루틴으로써 찾고 있는 화일에 대한 엑세스요청이 정당한 엑세스인가를 확인하고 허락여부를 결정하기 위해 ACM을 호출하게 된다. 따라서 엑세스에 대한 감시는 화일에 대한 엑세스를 위해 최종적으로 반드시 거쳐야 하는 dfdsrch에서 실시하고 엑세스허가 결정은 ACM에 의해서 이루어 진다. 이때 ACM의 결과가 false인 경우 에러를 반환하고 dsopen은 화일의 open을 중단하게 된다.

다. 암호제어

화일의 암호화처리는 첫째는 화일내 정보의 비밀성(security or privacy)을 보장하고 둘째 무결성(integrity)을 보장할 수 있다[Stei85]. 즉 불법적인 자료의 노출(disclosure)을 방지하고 변조(modification)를 탐지 가능하게 한다. 그러나 암복호화는 화일의 load나 저장시마다 빈번히 수행해야 하므로 시스템의 처리속도와 효율성이 떨어지게 된다. 따라서 꼭 필요한 자료에 한해서 선택적으로 암호화가 이루어지도록 하여야 한다.

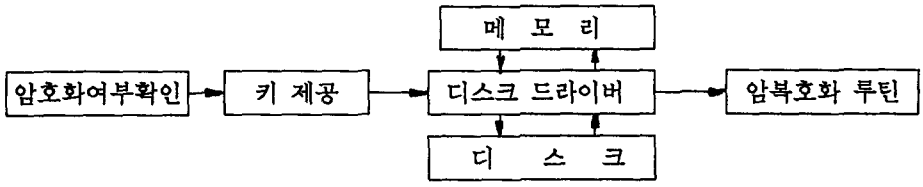
(1) 암호화 메카니즘의 설계 및 구현

화일의 암호제어는 시스템에서 엑세스 권한 확인이 끝난 다음에만 수행될 수 있는 디바이스 드라이버의 Control routine에서 이루어지도록 설계하였다. 즉 엑세스제어의 과정에서 엑세스권한을 획득하면 <그림 12>과 같이 화일의 암호화 여부를 확인하여 암호화화일인 경우 키를 제공하도록하여, 키를 제공하는 정당한 자에게만 메모리상에 화일의 내용을 로드시킬 때 암호화를 실시하고 복호화는 메모리에 있는 화일을 디스크에 저장시 이루어지도록 하였다.

화일의 암복호화처리를 위해서 화일헤더에 화일 암호화비트를 두어 암호화 여부를 나타낼

수 있도록 하였다. 그리고 암복호화를 위한 키의 제공은 화일 암호키를 마스터키(master key)로 암호화하여 화일별로 유지하고 개인고유키제어하에 시스템이 제공하도록 하였다. 개이고유키 키관리방법은 다음에 설명한다.

암복호화를 위한 알고리즘은 DES를 이용하여 crypt()루틴으로 구현하였으며, 암복호화 단위는 버퍼크기인 512바이트 단위로 암복호화 하도록 하여 메모리의 효율성을 기하였다. 암복호화 수행환경 루틴은 dsopen(), lfsflush(), lfsetup()에서 이루어지도록 하였다.



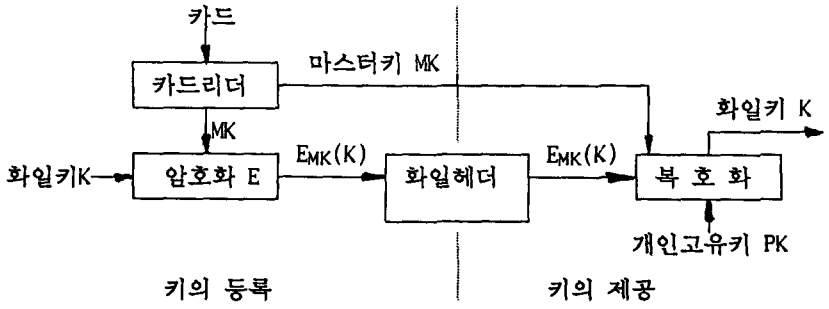
<그림 12> 암호화 메카니즘

(2) 개인고유키제어 키관리 메카니즘의 설계 및 구현

화일 보호를 위한 암호화 제어의 설계시 가장 중요한 고려사항은 키의 생성, 분배, 관리에 대한 문제이다. 암호 시스템의 보안성은 결국 키의 보안에 의해서 결정되므로[Stei85], 안전한 암호 시스템에서는 보안성있고 효과적인 키의 관리가 반드시 전제되어야 한다.

본 논문에서는 키관리를 위해 개인고유키제어 키관리의 개념을 설계 구현하여 제시하고자 한다. 개인고유키제어 키관리는 PC에 등록된 모든 소유자는 개인고유키를 갖도록하며, 암호화 서비스는 개인고유키를 시스템에 제출하면 시스템은 사전에 단방향 함수에 의해 등록된 개인고유키와 비교하여 인증하고, 인증된 사용자에 한하여 시스템이 암호화하여 저장시켜 놓은 화일 암호화키를 복호화하여 제공하게 하는 개념이다. 따라서 개인은 화일키를 직접 관리하지 않고도 키를 사용할 수 있도록하는 간접제어 기법이 된다.

<그림 13>에서 보는 바와 같이 시스템에 보관되는 화일 암호화키는 노출되지 않도록 마스터키(master key)에 의해 암호화되어 화일 헤더의 설정된 키필드(key field)에 저장되며 암호화 서비스시 마스터키에 의해 복호화 되어 제공된다. 최초 화일 암호화키의 등록은 화일 생성시 생성자에 의해 입력되며 입력 즉시 DES알고리즘으로 구현된 crypt() 루틴에 의해 암호화되어 저장된다. 이때 키의 길이는 DES알고리즘에서 사용될 수 있도록 8개의 문자열로 한다(개인고유키, 매스터키도 동일함).



<그림 13> 키관리 메카니즘 개념도

마스터키의 관리는 카드에 수록하여 시스템 매니저에 의해서 제공되도록하는 방법을 제안하며, 본 논문에서 실제 구현할 때는 카드리더를 사용하지 않고 소프트웨어적인 방법을 사0용하였다.

개인고유키의 생성관리는 패스워드와 같은 방법을 적용하며 개인이 변경 사용할 수 있도록 chpsk()를 구현 하였다.

개인고유키는 시스템관리자(SA)에 의해 테이블관리모듈(TMM)에서 등록되고 개인은 이를 변경하므로써 고유키가 생성된다. SA에 의해 등록되는 키는 단방향 함수에 의해 개인정보 테이블에 저장됨으로서 키테이블에 접근한 사람도 사용자의 개인고유키를 알아낼 수가 없다.

라. 보안 컴맨드의 구현

(1) copy()

copy의 기능은 A화일을 읽어 B화일에 기록하는 것이다. 이때 source인 A화일의 내용에 상관없이 목적화일인 B화일의 보안결정요소(비밀등급, 암호화여부등)에 의해 관리되기 때문에 원래의 보호수준의 보호를 받지 못하는 경우가 발생된다. 따라서 이를 방지하여 copy시 source화일의 보안요소가 그대로 목적 화일에 전달되어 똑같은 보호를 받을 수 있도록 하여야 한다.

본 논문에서는 화일의 내용이 복사되어도 소스화일과같은 수준의 엑세스제어가 가능하도록 하며, 암호화된 화일인 경우 암호화된 상태로 copy되어 불법적인 복사유출이 발생하여도 정당한 권한 없이는 화일을 이해할 수 없게 하였다.

이를 위해 copy() 루틴을 수정 구현하였으며 copy()루틴 외에도 dsopen(), dfdsrch()을 수정하였다.

(2) del()

화일을 삭제하는데 보안성을 고려하여 기존의 del 컴맨드를 수정하였다. 화일을 삭제하기 위한 컴멘드인 기존의 del은 화일의 인덱스블럭 및 데이타블럭의 포인터를 끊음으로써 화일을 삭제한다. 따라서 pctool등의 도구를 이용하여 이를 재생시킬 수가 있어 비밀의 누출 위험이 있다. 따라서 비밀이 들어 있는 화일을 삭제할 시는 회복 불가능하도록 하여야 한다.

본 논문에서는 삭제된 화일을 재생할 수 없도록 하기 위해서 del() 명령루틴 내에서 삭제할 화일 전체에 문자 '0'를 over write한 다음 포인터를 제거하고 인덱스블럭과 데이타블럭을 반납하도록 수정하였다. 또한 불법적으로 또는 실수로 비밀 화일을 삭제하는 경우를 방지하기 위하여 화일의 비밀 수준보다 사용자의 로그인 비밀등급이 높아야 하고, 화일의 화일의 소유자이어야만 화일을 삭제할 수 있도록 하였다.

마. 감사추적(audit trail)

효과적인 감사추적은 안전한 시스템의 주요한 구성요소로써 TCSEC에서도 accountability를 위한 요구사항으로 채택하고 있다.

감사추적의 일반적인 목적은 (1) 사용자나 프로세스들의 엑세스 기록이나 각 객체들에 대한 엑세스 형태를 점검할 수 있도록 하고, (2) 보호메카니즘을 우회하려는 불법적시도를 발견할 수 있도록 하며, (3) 사용자들의 불법 우회시도를 저지할 수 있도록 하는데 있다 [Glig85][Glbu86].

본 논문에서 감사추적기능을 다음과 같이 제안하고자 한다.

-보안위반에 대한 기록 및 report기능

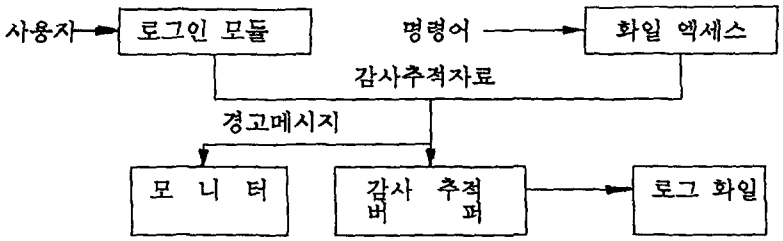
-보안 위반에 대한 경고기능

- log file의 보호

감사추적이 이루어져야 하는 시기와 장소는 다음과 같이 선정할 수 있다.

- 사용자의 시스템 엑세스를 위하여 로그인 할 때 사용자 ID, 비밀등급, 날자 및 시간, 실패여부를 기록.
- 사용자가 로그인한 후 화일에 대한 엑세스를 하고자 할때 사용자ID, 비밀 수준, 접근모드, 시간, 날자, 접근대상, 엑세스종류 등을 기록.

본 논문에서는 감사추적기록의 보호는 다른 화일과 분리해서 화일시스템으로는 접근할 수



<그림 14> 감사추적 메카니즘

없는 디스크블럭을 설정 기록함으로써 정상적인 화일 시스템을 통해서는 접근할 수 없도록 하고 시스템 프로세스만이 접근할 수 있도록 할 것을 제안 한다. 이와 같은 설계개념을 <그림 14>과 같은 메카니즘으로 표현할 수 있다. 즉 사용자의 로그인시와 화일 엑세스시 감사정보를 버퍼에 유지하여 로그화일에 기록하며 불법적인 시도가 반복될 때는 경고문을 모니터에 전시하여 경고하도록 하여야 한다.

본 논문에서는 감사추적기능의 구현은 생략하였다.

V. 결 론

본 논문에서는 보안성있는 PC보안시스템을 PC-XINU 운영체제의 커널수준에서 설계하고 구현하였다. 구현한 PC보안시스템은 다단계 보안메카니즘을 갖도록 하였으며, 정보흐름제어가 가능하며 군과 같은 특수한 비밀체계에서 적용될 수 있고, 개인의 소유권 또는 프라이

버시를 보호할 수 있는 엑세스제어정책을 적용하였다.

적용된 보안 메카니즘은 사용자 ID와 패스워드, 신원허가를 인증정보로 하는 신원확임 및 인증과정과 MAC정책과 DAC정책을 결합한 엑세스제어, DES를 이용한 암호제어와 개인고유키 제어 키관리 그리고 감사추적기능등이며 감사추적기능을 제외한 부분을 구현하였다.

본 논문에서 연구한 PC보안시스템은 범용성있는 PC 운영체제에 적용될 때 충분한 보안성을 제공할 수 있을 것이며, 감사추적 기능은 보안성있는 PC를 위해 필수적인 요소이므로 계속적인 연구가 요구된다.

참고문헌

[Cha90] 차승열, "PC-XINU 운영체제에서의 안전한 화일시스템 설계 및 구현", 국방대학원, 1990.

[CoFo88] Douglas Commer, Timothy Fossum,"Operating System Design Vol.2 the XINU Approach",Prentice Hall, 1988.

[Denn82] D. E. Denning, " Cryptography and Data Security ", Addison Wesley, 1982.

[HoLe90] 홍기용, 이철원, 이형수, 박태규, "MLS O.S를 위한 엑세스 제어 메카니즘연구", 한국전자통신 연구소, 제 2 회 정보보호와 암호에 관한 워크샵 논문집, 1990. 9.

[Land81] Carl E. Landwehr, "Formal Models for Computer Security", ACM Computing Survey, Vol.13, No.3, 1981.9, pp. 247-287.

[Mead83] Ft.George G. Mead, "Excerpts from DOD Trusted Computer System Evaluation Criteria", CSC-STD-D01-83, DOD Computer Security Centter, 1983.

[Nam 90] 남길현, "암호시스템의 특성과 활용", 정보과학회지, 1989, PP.55-64.

[NCS 85] NCSC(National Computer Security Center), "Personal Computer Security Consideration", NCSC Pub. WA-002-85, 1985.

[Pfle89] Charses P. Pfleeger, Security in Computing, Prantice Hall, 1989.

[SeMe90] Kemeth F. Seiden, and Jeffrey P. Melanson, "The Auditing Facility for a VMM Security Kernel", Proc. of IEEE Computer society Symposium on Research in Security and Privacy, 1990, PP. 262-277.

[Sin 89] 신장균, 최은재, "화일 보호와 안전한 운영체제", 정보과학회지, Vol.7, No.5, 1989.10.