USB 보안 모듈을 이용한 정보 보호 시스템 설계

조경옥* 김종우* 김영진** 한승조*

*조선대학교 전자정보통신공학부 **데이콤

Design of Information Protection System Using USB Security Module

Kyoung-ok Cho* Jong-woo Kim* Young-jin Kim** Seung-jo Han*

*School of Electronic and Information Communication Eng. Chosun Univ.

**DACOM

요 약

현 시대는 유형적인 물질 보다 무형적인 정보의 가치가 중요시되고 있다. 특히 소프트웨어의 불법 복제는 정보화 시대의 가장 큰 역기능으로 작용할 뿐 아니라 국가 신용도를 평가하는 중요한 항목으로 자리 매김하고 있다. 그러나 기존의 상용화된 소프트웨어 불법복제 방지 제품들은 복제방지 기술이 미약하여 쉽게 락이 크랙 되어 복제방지의 기능을 발휘 할 수 없는 것들이 대부분을 차지하고 있다. 본 논문에서 제안하는 복제방지 전용 칩을 사용하여 하나의 락으로 여러 소프트웨어를 동시에 보호하는 기능뿐만 아니라 IBM PC 호환기종의 USB 인터페이스를 가지는 정보 보안 모듈의 설계한다.

I. 서론

현대사회는 정치, 경제, 행정, 문화 등 인류가 더불어 살아가면서 형성된 모든 영역이 정보통신 시스템을 통해 생성되고 유포되는 지식정보 기반의 사회구조로 급속히 진행되고 있고 정보화 시대가 현실화되면서 개인의 사생활이나 개인정보는 중요한 이슈가 되고 있는 실정이다.

또한 최근 인터넷에 대한 관심의 증대와 인터넷에 연결된 호스트의 숫자가 폭발적으로 증가함에 따라 사용자는 인터넷에 접속하여 다양한 형태의 정보 및 여러 종류의 통신 서비스 등을 쉽게 제공받고 있다. 반면에 인터넷 망을 통해 불법사용자 및 해커의 침입 등으로 정보의 손실, 파괴, 변조 등에 의한 피해가 늘고 있다.

현재 개발되어지고 있는 소프트웨어 및 개인정보의 보안시스템은 소프트웨어나 하드웨어로 구현하게 되는데, 보안기능을 탑재한 소프트웨어는 보안모듈과 정해진 방법으로 통신함으로써 모듈의 유무나 오류 등을 파악하게 된다. 보안 강도로는 소프트웨어로 이루어진 장치보다는 하드웨어로 제작된 보안모듈의 강도가 높다. 그러나 하드웨어 보안모듈은 사용되는 소프트웨어 제품의 가격에 큰 영향을 끼칠 정도로 고가이기 때문에 고가의 소프트웨어에만 적용되고 있다. 또한 기존의 상용 하드웨어 보안모듈은 사용상의 불편함과 기능의 한계성을 가지고 있어서 소프트웨어 보호 측면에서 보면 개발업자들에게는 환영을 받지만 소프트웨어를 사용하는 최종사용자 입장에서는 외면 받기 십상이었다. 따라서 소프트웨어 복제방지 및 정보보호를 위해서는 사용이 편리하고 저가이며, 보안 강도가 높고 역기능 방지에 강력한 암호알고리즘을 가진 견고한 하드웨어 보안모듈이 필요한 실정이다[5].

본 논문에서는 복제방지 전용 칩을 이용하여 소프트웨어의 복제 장지 기능 및 하나의 락으로 여러 소프트웨어를 동시에 보호하는 기능, PC보안 기능, Web Browser 인증 기능을 내포하고 있는 IBM PC 호환기종의 USB 인터페이스를 가지는 보안 모듈을 설계하고자 한다.

II. 복제방지 및 보안장치

현재 보편화되어 있는 보안장치의 동작 형태를 보면 그림 1과 같다. 그림 1의 (a)에서 보여진 초기의 단순형 보안장치는 단순히 물리적인 보안모듈이 존재하는 것만을 검사하는 것으로써 보안모듈로 전송하는 데이터나 보안모듈로부터 반환되는 데이터의 형태가 불변이다. 두 번째 그림 1의 (b)는 초기의 보안장치가 쉽게 에뮬레이션이 가능한 점을 보완하기 위하여 정적인 제어가 아닌 동적인 제어를 하도록 하는 데이터 형태가 변형되는 지능형 보안모듈의 구조가 있다[2][5].

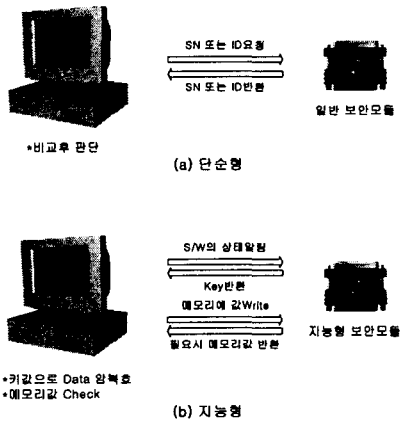


그림 1 일반적인 보안장치의 동작 형태

이와 같은 보안장치의 검사 알고리즘은 보다 복잡한 제어소프트웨어의 구조를 요구하게 되며, 수행에 따른 시스템의 부하를 유발하는 특징을 갖는다.

1. 보안장치의 동작 방식에 따른 특징

1) 소프트웨어 보안장치

- 물리적인 특성을 이용한 경우
- 사용자 등록제를 이용하는 경우
- Time Stamp를 이용하는 경우

2) 하드웨어 보안장치

- 보안모듈의 유무를 검사하는 방법
- 보안모듈에 정보를 저장하는 방법
- 보안모듈을 이용하여 원본을 변형하는 방법 [3][4].

2. 보안 모듈설계의 특징

복제방지 시스템은 특성상 하드웨어에 의존적이다. 따라서 구현에 따른 수행시간이 많이 소요되는 부분은 하드웨어로 설계하고 수행시간을 고려하지 않아도 되는 부분은 소프트웨어로 설계하여야만 한다. 하드웨어로 구현함으로써 얻게된 장점은 다음과 같다.

- 데이터가 병렬처리가 가능
- 수행속도를 최대화
- 크랙이나 해킹이 불가능

III. 보안모듈 설계

1. USB 인터페이스 설계

1) USB 컨트롤러

USB는 2개의 신호선을 통한 시리얼통신을 하도록 규정되어 있으며, USB 컨트롤러내부에서 버퍼링 후 데이터의 처리를 수행하게 된다. USB 컨트롤러는 8bit 프로세서 및 EPROM을 내장하고 있으며, 32바이트의 입출력 버퍼를 이용하도록 설계되었다.

그림 2는 USB 인터페이스 구성 및 컨트롤러의 블록도이다.

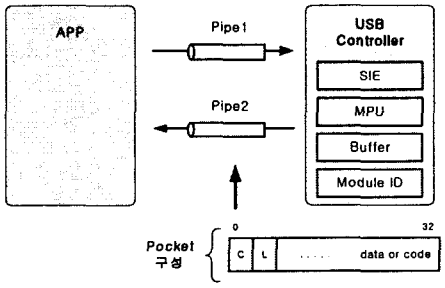


그림 2 USB 인터페이스

본 논문에서는 Cypress사의 USB 내장 마이크로 컨트롤러(CY7C64013)를 사용하여 PC와 통신을 한다.

CY7C64013의 spec은 다음과 같다.

- 8Bit Harvard architecture Microcontroller
- 256 byte of RAM and 8KB of EPROM
- USB Specification Compliance
- Integrated USB transceivers

가. 전기적 특징

USB에서는 한 쌍의 차동신호(D+/D-)를 사용해서 데이터 신호를 실행하는 것이지만 이들 D+/D-를 스윙시킨 전압레벨(DC 특징)은 TTL 레벨보다 다소 엄밀한 값이 요구된다.

USB에서는 Full 스피드인 12Mbps의 전송 스피드를 내는 전송모드와 로우 스피드인 1.5Mbps의 전송 스피드를 내는 전송모드가 있는데 그들의 타이밍 특성은 각각 다르다.

로우 스피드의 Function은 기본적으로 휴먼 인풋 디바이스(HID) 등의 소형 주변기기에 쓰이게 되어 저렴한 가격으로 구성될 가능성이 크다. 본 논문에서는 대량의 데이터를 암복화 해야 하기 때문에 Full 스피드를 채택했다.

㉠ 디바이스의 전송속도

Low Speed 와 Full Speed의 구별은 USB 디바이스의 상위 D+ 와 D- 중에 어느 쪽에 Full Up 저항을 연결하느냐에 따라 결정된다. Full Speed는 D+ 에 Low Speed는 D- 이다.

㉡ Frame에 의한 시간분할

USB의 데이터 전송은 단위가 Frame으로 시간분할방식이고, 그 Frame을 겹쳐 쌓아 가는 것으로 USB의 데이터 전송방식에 있어서는 중요한 개념이다. USB는 Frame을 1ms의 시간단위로 반복된다. Frame은 SOF(Start Of Frame)의 패킷에 의해 시작된다. Host PC는 미리 그 Frame 가운데 스케줄링 된 데이터 전송요구 토큰을 차례대로 송출함과 동시에 여러 Function과의 데이터전송을 병행해 나간다. 그리고 전송할 데이터가 큰 경우는 한 Frame에 보낼 수 없으므로 다음 Frame에 다시 데이터 전송 요구 토큰을 PC에서 그 Function에 송출하는 것으로 데이터 전송을 완료시킨다. 대량 데이터의 경우는 데이터 열을 Frame 단위로 분할해서 전송해야만 한다. 따라서 USB Function에는 프레임 단위의 데이터 열을 저장하는 FIFO가 있어야한다. 1회의 Frame에 전송할 수 있는 데이터 량은 전송방식에 따라 달라지지만 1~1,023 byte이다.

㉢ 데이터 전송방식

USB의 데이터 전송방식은 총 4가지이다. Isochronous, Interrupt, Control 그리고 Bulk 전송이 있다.

Isochronous 전송은 일정 주기로 일정 양의 데이터를 전송하고 싶을 때, 바꿔 말하면 데이터의 실시간 전송이 요구되는 데이터의 경우에 사용된다. 예를 들면 전화기의 스피커나 마이크로폰의 음성데이터의 전송, 또는 카메라의 영상신호 전송 등 전송 중에 에러가 발생하여도 지장이 없는 멀티미디어 장비에 주로 사용된다.

Isochronous 전송의 특징을 크게 세 가지로 나누면

- 데이터의 전송폭을 보장하고 있다.
- 데이터의 전송시간을 보장하고 있다.
- 데이터의 에러 보장은 하지 않는다.

Interrupt 전송은 Function에서 포스트 PC에 주기적으로 소량의 데이터를 입력하는 애플리케이션으로 사용된다. 예로 키보드 그리고 마우스, 조이스틱이 대표적인 예이다. 이 방식은 단방향 전송이다.

Control 전송은 USB 디바이스가 그 Configuration 정보 등의 호스트 PC에 전송 할 때 사용된다.

Bulk 전송은 대량의 데이터를 전송하고, 데이터가 부정기적으로 발생하고, 리얼타임 성이 그다지 중요하지 않을 때 사용된다. Isochronous와는 달리, 데이터 패킷에서 에러가 발견되었을 경우에는 재송요구가 가능하므로, 최종적으로 전송되는 데이터의 내용은 보장된다.

㉣ USB 패킷 Protocols

각각의 USB transaction은 몇 가지 부분으로 나누어지는데 있는데 아래와 같다.

- Token 부, 디바이스의 주소를 포함하고, 특정한 디바이스로의 전송을 하는 정보들을 포함하고 있다.

- Data Packet 부, 데이터 전송이 요구될 때만 사용된다.
- Handshake Packet 부, 에러체크와 재전송 등의 정보를 포함하고있다.

㉤ USB Descriptor 구성

Descriptors 란, USB 장치의 특성을 정의해 놓은 것이다. descriptor는 아주 특정한 형식으로 구성된 binary 데이터 구조이다. 각각의 descriptor의 시작은 descriptor의 전체 바이트 수를 나타낸다 (한계는255 바이트). 그 다음의 1 바이트는 descriptor의 타입을 나타낸다. 아래에 descriptor 들의 계층도를 나타내었다.

- device descriptor
- configuration descriptor
- interface descriptor

사용되는 Endpoint 수는 4개이며, class 와 sub-class, protocol은 정해지지 않은 Mode를 사용한다. 이 인터페이스의 string은 5개로 설정되어 있다.

- class descriptor

Hid 장치에만 사용된다. USB 보안모듈에는 해당사항 없음.

- endpoint descriptor

4개의 Endpoint를 사용한다. USB 장치의 Endpoint 1 과 2, 두 개를 사용하나 1번의 In 과 Out 2번의 In 과 Out 총 4개의 Endpoint 가 된다. 그러나, 이 시스템에서는 Endpoint 1번의 Out 과 Endpoint 2번의 In 만을 사용한다. (Pipe 0,1,2,3 중에 1과 2만 사용.)

USB-spec에 따르면 transaction에 최대 64 byte 까지 전송할 수 있으나, cy7c6401x 은 최대 32 byte이다.

- string descriptor

USB 장치의 설명을 기입하는 곳(사용하지 않아도 됨).

스트링은 UNICODE로 정의됨.

2. 패킷 구성

패킷은 총 32 byte로 구성되어있으며 입력패킷과 출력패킷 두 가지로 구성되어있다. 출력패킷은 Endpoint 1을 사용하며, 암호칩 초기화, Fake Master Code, 기타코드, 데이터전송, User String 요구 등의 명령을 수행할 수 있다. 입력패킷은 Endpoint 2를 사용하며, 복호화 된 데이터 또는 User String을 받을 때 사용된다.

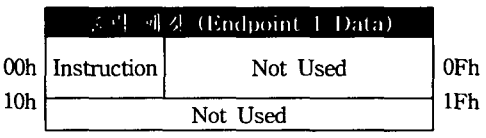
■ 명령별 패킷의 구성

- 암호칩 초기화
- 여러 종류의 Code를 암호칩으로 전송
- 데이터를 암호칩으로 전송
- 모듈정보 요구
- 모듈내부의 Protector Code를 암호칩으로 전송

1) 암호칩 초기화

암호칩을 초기화시키는 기능을 수행한다. 패킷 구성은 Instruction 만 존재한다.

Instruction = 0x01h (1byte)



2) 여러종류의 Code를 암호칩으로 전송

암호칩에 Fake Protector Code와 기타 여러 종류의 코드값을 전송한다.

Instruction = 02h(1byte)

Code Len = 0Ch(13byte) or 11h(17byte) (컨트롤워드 포함)

Code = Protector_Code, External_Code, External_Dec_Code (16byte),

Serial_Code, Random_Code (12byte)

Control Word(CW)(1byte)

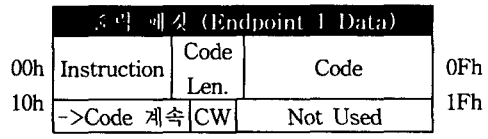
Protector Code = 40h

Serial Code = 20h

Random Code = 10h

External Code = 08h

External Dec Code = 04h



3) 데이터를 암호칩으로 전송

암호화된 실행파일의 Code부분을 Pipe1을 사용하여 모듈로 전송한 후 Pipe2를 통하여 복호화된 데이터를 읽어들인다.

* 출력패킷

Instruction = 04h(1byte)

Code Len = 0Ch(1byte)

Data = 원본데이터, 암호화된 데이터나(12byte)

Control Word(CW)(1byte)

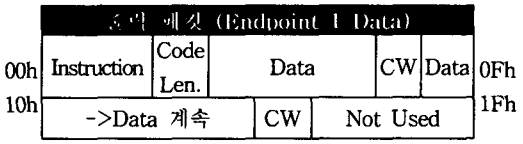
실행파일 Data = C0h

내부키 사용 암호화 = A0h

외부키 사용 암호화 = 90h

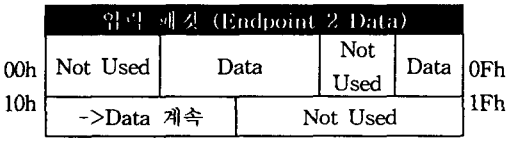
내부키 사용 복호화 = 88h

외부키 사용 복호화 = 04h



* 입력패킷

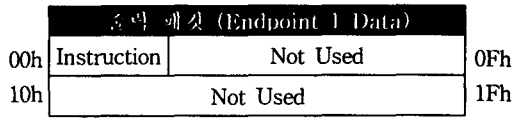
입력 데이터는 12byte로 주소(02h~0Dh)와 주소(0Fh~1Ah) 이다.



4) 모듈정보 요구

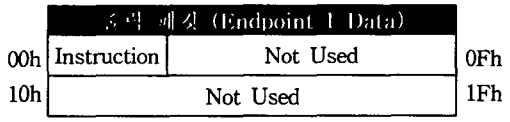
USB 모듈 내에 정의된 정보를 요구한다. 각각의 모듈마다 다른 값을 가지고 있음.

Instruction = 08h(1byte)



5) 모듈 내부의 Protector Code를 암호칩으로 전송

Instruction = 10h(1byte)



6) 모듈 내부의 Serial EEPROM에 값을 기록

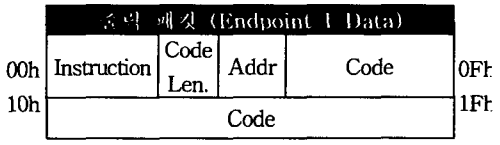
Instruction = 20h(1byte)

Code Len = 어드레스를 포함한 코드길이(1byte)

Address = 기록하고자 하는 값의 시작 주소(1byte)

Code = 입력할 코드(임의의 길이)

다른 명령과는 달리 여기서는 컨트롤 워드가 필요 없다.



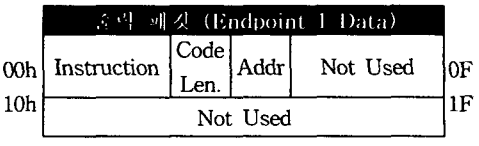
7) 모듈내부의 Serial EEPROM에서 값을 읽는다.

* 출력패킷

Instruction = 40h(1byte)

Code Len = 읽고자하는 코드의 길이(1byte)

Address = 읽고자하는 코드의 주소(1byte)



* 입력패킷

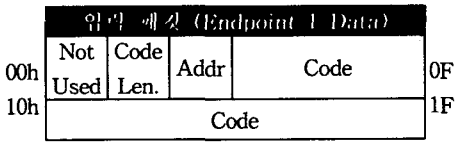
입력 데이터는 12byte로 주소(02h~0Dh)와 주소(0Fh~1Ah) 이다.

Not Used = 쓰레기 값(2byte)

Code Len. = Address 포함한 읽혀진 코드의 길이(1byte)

Address = 읽혀진 값의 주소(1byte)

Code = 읽혀진 값(임의의 길이)



3. 보안모듈 구성

보안모듈은 소프트웨어의 보호 및 개인정보 보호시스템에 사용되는 모듈로서 다양한 인터페이스에 채택될 수 있다. 본 논문에서는 USB 인터페이스를 채택함으로써 범용화되어 있는 USB포트에 PNP(Plug and Play)기능을 갖춤으로써 즉시 사용가능한 보안장치로 설계하였다. 보안모듈은 크게 암호칩, USB 컨트롤러 그리고 EEPROM으로 구성하였고 그림 3과 같은 구조로 되어있다.

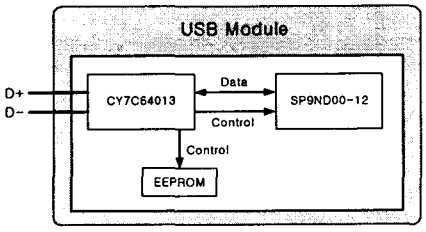


그림 3 보안모듈의 구성

USB 컨트롤러는 USB Spec. 2.0의 고속에 맞춰 설계하며, USB Interface 엔진, USB 송수신장치, 8bit RISC 마이크로 컨트롤러, Clock Oscillator, 타이머 그리고 프로그램 메모리 등을 내장한다. 또한 USB 컨트롤러는 USB 통신과 함께 암호칩과 EEPROM의 컨트롤을 수행하고, 실행파일, 데이터파일, 소프트웨어 등록정보 및 사용자 정보 등을 Flash Memory를 통하여 관리하게 한다.

암호칩은 D-DES 블록, Hash 블록, Scrambler 블록으로 구성되며 데이터의 암·복호화 및 변경 검사와 스크램블링을 수행한다.

Ⅳ. USB를 사용한 보안 모듈 시뮬레이션

1. 시뮬레이션 환경

- 운영체제 : Windows 2000 Server, XP
- 개발언어 : Visual C++6.0, Visual Basic 6.0
- 보 안 칩 : CU9NO00-12(자체 개발칩)
- 컨트롤러 : Cypress사의 USB 내장 마이크로 컨트롤러(CY7C64013)
- 퍼스널컴퓨터 : 펜티엄-4

2. 통합시스템 제작

본 논문에서 제안한 보안모듈의 시뮬레이션 한 결과는 보안모듈을 통한 SW 복제방지를 실현하기 위한 개발도구로서 자동 구현과 수동구현을 지원한다. 자동구현은 소스레벨이 아닌 실행코드레벨에서 적용이 가능한 구현으로 SW 개발자가 개발 완료한 SW의 실행코드에 직접 보호알고리즘을 적용함으로써, 사용상의 편의성과 보안기술 적용기간을 단축할 수 있다. 반면에 소스레벨의 적용을 위한 수동구현은 라이브러리 형태의 보호기능 함수들을 SW 개발사의 소스에 첨가하여 컴파일 함으로써 DK(개발 키트)를 이용한 보다 다양한 서비스가 가능한 장점을 갖는다.

그림 4는 보안 모듈을 이용한 DK의 실행 화면이다.

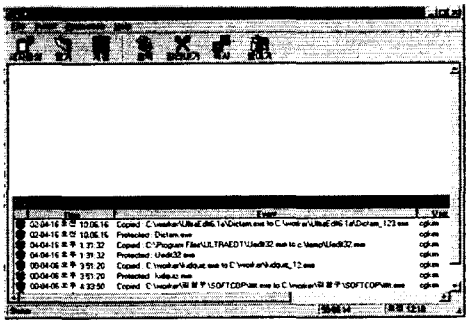


그림 4 보안 모듈을 이용한 DK의 실행화면

보안 모듈을 이용한 DK는 일반적인 SW 외에

Shareware나 Freeware의 SW에 적용하는 것도 가능하도록 설계하여 Shareware나 Freeware의 SW의 경우 제한사항이 쉽게 크랙 되어 정품처럼 유통되어지는 방법을 차단할 수 있다.

그림 5는 자동 구현을 위한 실행제한 옵션설정 창이다.

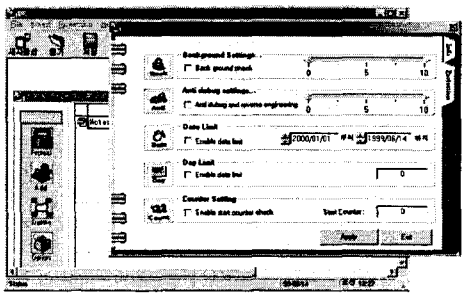


그림 5 DK의 실행제한 옵션 설정

가. DK-Manager

DK-Manager는 외부적으로는 보안모듈의 사용자와의 인터페이스를 담당하고, 내부적으로는 SW 관리 드라이버와 PC 관리드라이버 그리고 응용프로그램간의 통신을 처리하는 중추적 역할을 한다.

제작된 DK-Manager가 그림 6에 나타나 있다.

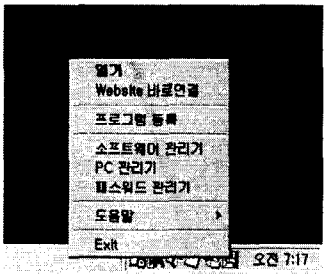


그림 6 DK-Manager

DK-Manager는 Tray Icon 형태로 상주하게되며 그림 6에서와 같은 메뉴를 갖으며, 세부적으로 다음의 기능을 갖는다.

- S-Cop 보안모듈의 관리
- S-Cop 보안모듈의 로그관리
- SW의 등록관리
- SW의 사용인허가 관리
- PC 관리를 위한 설정 처리
- 사용자별 패스워드 관리
- 사용자별 PC 환경 관리
- Virus 탐지

나. SW Manager

SW Manager는 실행을 요청한 응용프로그램에 대해 인증 Token을 SW 관리 드라이버로부터 발급 받아 넘겨주고, 응용프로그램이 종료될 때까지 계속 모니터링을 통해 관리하는 역할을 하게 된다. 만약 실행 중에 어떠한 이벤트(예로 보안모듈의 제거시)가 발생 시 약정된 방식에 따라 프로그램에 제한을 가하게 된다. 이때 프로그램의 제한에 대한 정의는 응용프로그램의 개발자에 의해 정해지며, 프로그램의 강제종료에서부터 프로그램에 대한 메시지 전송까지 가능하다.

그림 7은 SW 관리자의 실행창이다.

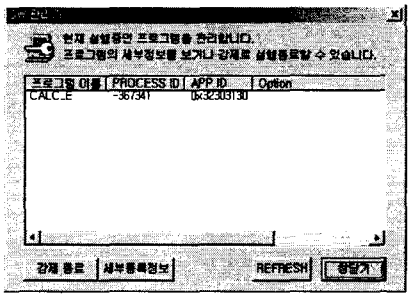


그림 7 SW 관리자의 실행 창

다. PC Manager

PC Manager는 PC에 대한 보안 설정을 담당하며, 설정된 값은 PC 관리 드라이버에 전송되어져 보안 기능이 활성화되어진다.

그림 8은 PC 관리기의 실행창이다. 그림에서 파일 보안은 모듈에 의한 데이터의 암·복호화를 자동으로 처리하기 위한 항목이며, 특정기능 차단은 컴퓨터의 특수한 기능에 대한 접근을 통제하기 위한 기능이다. 특정기능 차단은 공용 컴퓨터 등의 시스템 설정 변경 방지 등에 이용되어질 수 있다. 그리고 특정단어 입력 차단은 키보드를 통해 입력되는 데이터 중 특별히 지정된 단어에 대한 제 3자의 억세스를 거부하는 기능으로 음란싸이트

차단 등으로 이용되어 질 수 있다.

현재 PC 보안에 사용되는 암호 알고리즘은 SEED, SPE-128 그리고 Triple DES가 있으며 키는 128bit~192 bit가 사용된다.

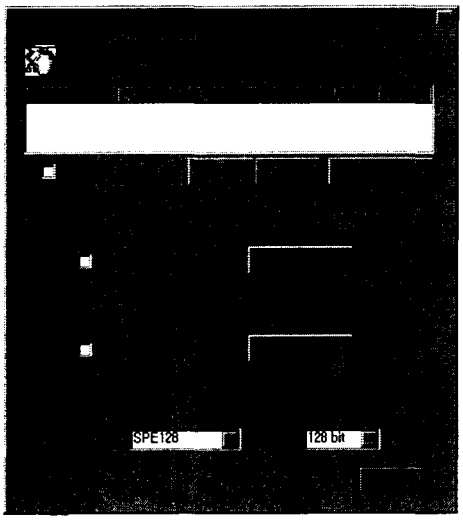


그림 8 PC 관리기의 실행

V. 결 론

본 논문에서는 급속히 증가하는 프로그램 및 정보의 불법복제의 방지를 위한 암호화 보안모듈(락)를 구현하기 위하여 보다 안정성 있고 빠른 속도의 암호화 알고리즘을 도출하고, 하드웨어로 제작하게 될 때 얻게 되는 장점을 최대한 고려하여 암호화부 및 콘트롤부 등을 ASIC으로 구현한 암호칩을 사용하였다. 또한 락과 함께 락 제어용 드라이버 및 라이브러리를 설계하였다.

이와 같이 암호칩을 이용한 보안모듈은 소프트웨어의 보호 및 개인정보 보호시스템에 사용되는 모듈로서 다양한 인터페이스로 적용할 수 있으며, 본 논문에서는 USB 인터페이스를 채택함으로써 범용화되어 있는 USB포트에 즉시 사용 가능한 보안장치로 설계하였다.

이러한 결과는 개인정보와 컴퓨터 프로그램에 대한 보호가 시급한 현시점에서 보안모듈을 통하여 기밀성이 요구되는 개인정보의 보호와 소프트웨어의 불법복제방지를 가능하게 하였으며, 전자상거래나 전자화폐의 활용이 많은 최근 위·변조의 방지를 위한 개인용 보안장치, 인터넷 뱅킹의 인증서 저장장치, 전자상거래 인증장치 등으로 활용할 수 있을 것이며, 또한 급속히 초고속화 되어가고 있는 컴퓨터 통신망에서 비인가자에 대한 정보전송의 보호에도 효과적으로 활용될 수 있을 것이다.

참고문헌

[1] L. Brown, J. Pieprzyk, and J. Seberry, "LOKI- A Cryptographic Primitive for Authentication and Secrecy", Proc. of AUSCRYPT '90, pp.222-228. Jan. 1990.

[2] A. Herzberg and S. S. Pinter, "Public protection of software", In Adv- ances in Cryptology, Proc Crypto 85. H. C. Williams, Ed., pp.158, 1986.

[3] NBS, "Data Encryption Standard", FIPS Pub. 46, U.S, National Bureau of Standards, Washington DC, Jan. 1977.

[4] David Aucsmith, "Tamper resistant software", Information Hiding- Proceedings of the First International Workshop, pp.317-333, 1996.

[5] G. B. Purdy, G. J. Simmons, and J. A. Studier, A software protection scheme. In Proc. 1982 Symp. Security and Privacy, Oakland. CA, pp.99-101. Apr. 1982.

[6] M. G. Arnold and Mark D. Winkel, "Computer systems to inhibit unauthorized copying, unauthorized usage, and automated cracking of protected software", U.S. Patent No.4 558 176. issued Dec. 1985.

[7] D. E. Denning, "Cryptography and Data Security", Addison-Wesley, 1983.

[8] 윤용정, 공헌택, 남길현, "80비트블럭암호알고리즘(80-DES)의 설계 및 비도 분석에 관한 연구", 통신정보보호학회 논문지, 제5권, 제1호, 25-36, 1995. 3.

[9] Myung Shin Oh et al., "The Software Illegal Copy Protection using the Secure Chip", 2002 International Conference on Optical Communications and Multimedia, Chosun University, pp.50-54, Nov. 2002.

* 본 연구보고서는 정보통신부 정보통신연 진흥원에서 지원하고 있는 정보통신 기초연구 지원사업의 연구결과입니다.(University fundamental Research Program supported by Ministry of Information & Communication in republic of Korea)