스마트카드 기반의 효율적인 해킹 방지 시스템 설계

황 선 태* · 박 종 선**

Design of Efficient Hacking Prevention Systems Using a Smart Card

Suntae Hwang* · Jongsun Park**

Abstract

This paper describes the design of hacking prevention systems using a smart card. It consists of two parts, i.e., PC authentication and Keyboard-buffer hacking prevention.

PC authentication function is a procedure to handle the access control to the target PC. The card's serial number is used for PIN(Personal Identification Number) and is converted into hash-code by SHA-1 hash-function to verify the valid users. The Keyboard-buffer hacking prevention function converts the scan codes into the encoded forms using RSA algorithm on the Java Card, and puts them into the keyboard-buffer to protect from illegal hacking. The encoded information in the buffer is again decoded by the RSA algorithm and displayed on the screen.

In this paper, we use RSA_PKCS#1 algorithm for encoding and decoding. The reason using RSA technique instead of DES or Triple-DES is for the expansion to multi-functions in the future on PKI. Moreover, in the ubiquitous computing environment, this smart card security system can be used to protect the private information from the illegal attack in any computing device anywhere. Therefore, our security system can protect PC user's information more efficiently and guarantee a legal PC access authority against any illegal attack in a very convenient way.

Keywords : Smart Card, Hacking, Keyboard-buffer, Scan Code

논문접수일 : 2004년 2월 15일 논문게재확정일 : 2004년 5월 31일

* 대전대학교 정보통신인터넷공학부 교수
** 대전대학교 한방병원 전산실

1. 서 론

인터넷의 활성화에 따라 스마트카드(Smart Card)의 사용이 전 세계적으로 급증하고 있으며, 이와 관련한 기술 개발이 활발히 이루어지고 있다. 개인용 컴퓨터나 금융망, 행정망 및 의료망, 전자상거래(Electronic Commerce)등에서 스마트카드의 사용이 필수적으로 대두되고 있다[1]. 특히 앞으로의 정보통신 분야는 유비쿼터스 컴퓨팅(Ubiquitous Computing)환경으로 바뀌어 가고 있는 추세이다. 따라서 이와 관련된 기술과 제품이 점차 확대되어 가고 있으며, 이에 맞는 보안 시스템 환경이 구축되고 개발되어야 한다. 스마트카드를 이용한 개인의 보안 환경 제공은 이러한 유비쿼터스 컴퓨팅 환경에서 안전하게 시스템을 이용할 수 있게 해주리라 예측된다.

모든 인터넷 기반 환경에서 개인의 정보는 바이러스(Virus)와 해킹에 의해 항상 노출될 수 있다. 특히 전자상거래 및 인터넷 뱅킹이나 증권거래와 같은 전자 거래는 온라인으로 수행된다는 특성을 가지고 있으므로 보다 안전한 보안 메커니즘을 필요로 한다. 이와 같은 상황에서 최근 키보드버퍼 해킹(Keyboard-buffer Hacking)으로 인한 개인 정보의 노출 가능성이 점차 증대되고 있다. 키보드 버퍼 해킹은 키보드를 통해 PC로 정보가 입력되는 과정에서 키보드 버퍼에 임시 저장되는 ID, 패스워드, 신용카드번호 등을 빼 내가는 것을 말한다. 이러한 경우 PC사용자는 해킹으로부터 완전히 노출된 상태가 되며, 이것을 보호할 방법이 없게 된다. PC사용자의 정보 노출은 바이러스나 해킹에 의한 것뿐만 아니라 제3자를 통한 PC 접근을 통해서도 가능하다. 제3자의 PC접근 방지 기능은 현재 PC의 패스워드를 통해 이루어지고 있는 상태이지만, 이러한 문제를 해결하기 위해 효율적인 방안의 제시가 필요한 상황이다. 현재 스마트카드를 사용한 PC인증 기술이 점차 증가 추세에 있으며 특히, 스마트카드의 연구와 개발로 인해 인터넷상이나 사이버 공간에서의 새로운 보안 수단으로의 스마트카드 사용이 증가하는 추세이다. 스마트카드를 사용한 PC인증 기술은 지문인식과 같은 생체인증 기술과 함께 접목됨으로써 앞으로의 보안 인증 수단으로 크게 각광받을 것으로 사료된다.

본 논문에서는 최근 키보드 버퍼 해킹과 PC 사용자의 정보보호와 관련하여 키보드로부터 입력되는 모든 정보를 자바카드 플랫폼 상에서 암호화함으로써 해킹으로부터 정보 노출을 방지하고, 접근 권한을 가진 PC사용자만이 PC를 사용할 수 있는 효율적인 PC인증 시스템과 키보드 버퍼 해킹 방지 시스템을 설계하였다. 또한 이 시스템은 차세대 유비쿼터스 컴퓨팅 환경에서도 정보를 어느 곳에서나 안전하고 편리하게 처리할 수 있는 기능을 제공한다.

2. 스마트카드(Smart Card)의 특징

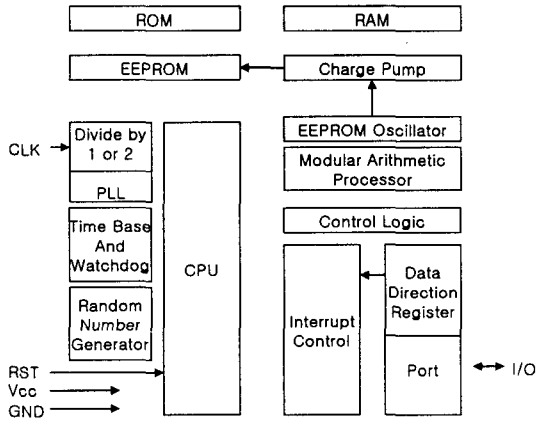
스마트카드(Smart Card)라는 용어는 그 적용 범위에 따라 다양하게 사용되지만 일반적인 스마트카드(Smart Card)에 대한 정의는 "마이크로프로세서, 카드운영체제, 보안모듈, 메모리 등을 갖춤으로써 특정 트랜잭션을 처리할 수 있는 능력을 가진 집적회로 칩(Integrated Circuit Chip)을 내장한 신용카드 크기의 플라스틱 카드"로 정의된다.

스마트카드는 ISO/IEC에서 규정한 IC 카드의 물리적인 구조와 인터페이스를 따르고 있다. ISO/IEC 7816-1에서는 접촉형 IC 카드의

물리적 성격인 카드의 형태, 크기 등을 규정하고, ISO/IEC 7816-2에서는 접점의 크기, 개수, 위치 및 기능을 규정하고, ISO/IEC 7816-3에서는 IC 카드와 단말기(IFD)간의 전기 신호 특성과 프로토콜 등을 규정하고, ISO/IEC 7816-4에서는 카드와 IFD간의 통신에 필요한 데이터 구조, 카드내의 파일 구성, 보안 체계 등을 규정하고 있다[2-6].

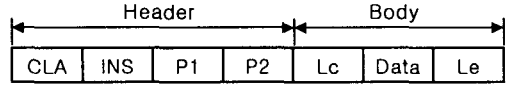
2.1 스마트카드의 H/W 및 명령어 구조

스마트카드 H/W 구조는 그림 1과 같으며, 5개의 활성화된 접점이 존재한다. CLK는 단말기로부터 제공되는 Clock 신호이고, RST는 Reset 신호이며, Vcc와 GND는 각각 공급되는 전압과 접지를 나타내고, I/O는 데이터 입출력이다. 스마트카드 H/W를 구성하는 주요 요소로는 CPU와 기억장치인 ROM, EEPROM, RAM, 그리고 그 외 논리회로들과 Charge Pump 등이 있다. 스마트카드와 단말기 간의 통신은 접속 단자들을 통하여 이루어진다[2, 3].

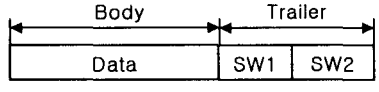


(그림 1) 스마트카드 H/W 구조

스마트카드 운영체제는 명령어와 응답 메시지를 사용하여 단말기와 스마트카드 사이에 통신을 한다. 스마트카드와 단말기 사이에 송신되는 명령어와 응답 메시지는 APDU(Application Protocol Data Unit)의 구조를 따르고, 그림 2와 같다[2, 6]. 여기서 CLA는 1-byte이고, Class byte이다. INS는 1-byte이고, 명령어 구분코드이다. P1, P2는 1-byte씩이고, 명령어에서 사용되는 변수 1과 변수 2를 나타낸다. Lc는 1-byte 또는 3-bytes이고, 송신 데이터 바이트 수를 나타낸다. Le는 3-bytes이고, 수신 데이터 바이트 수를 나타낸다. Data는 송/수신 데이터를 나타낸다. SW1, SW2는 1-byte이고, 응답 메시지에서 사용하는 상황 표시 1, 상황 표시 2를 나타낸다.



(a) 명령어



(b) 응답 메시지

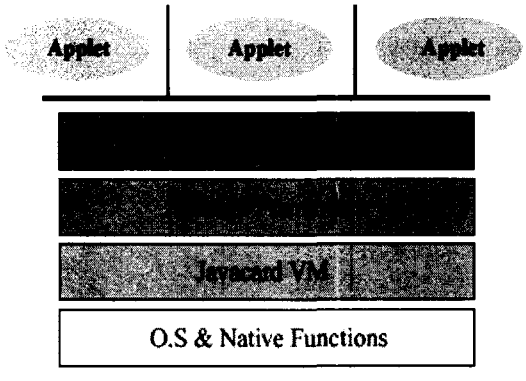
(그림 2) 명령어와 응답 메시지 구조

2.2 자바카드(Java Card)

자바카드란 자바로 작성된 프로그램이 수행될 수 있는 스마트카드이다. 일반적으로 자바카드는 그림 3과 같은 시스템 구조를 가진다. 각각의 스마트카드들은 카드마다 다른 종류의 하드웨어와 이러한 하드웨어를 운영하는 운영체제인 COS(Card Operating System)를 가지게 된다. 그러나 그 위에 자바카드 가상머신(Java Card Virtual Machine: JCVM)이라는 하나의 공통된 환경을 구현함으로써 한번 작성된 어플리케이션은 어떠한 스마트카드에서도

작동할 수 있는 하드웨어 플랫폼에 독립적인 자바카드 기술로 구현되어진다.

자바카드 가상머신 위에는 자바카드 프레임워크 및 기타 클래스 라이브러리들이 추가될 수 있다. 주로 이러한 프레임워크들과 클래스 라이브러리들은 기본적으로 지금까지의 스마트카드와의 호환성을 유지하면서 자바카드 기술을 이용하여 스마트카드 어플리케이션을 개발하기 위해서 필요한 각종 라이브러리들을 제공하게 된다. 또한 그림 3에서처럼 한 장의 자바카드에는 여러 개의 어플리케이션이 존재할 수 있다. 따라서 한 장의 자바카드가 여러 가지 기능을 수행할 수 있게 함으로써 스마트카드의 활용 범위를 넓혀줄 수 있다[7, 8]. 본 논문에서는 이러한 특성을 갖는 자바카드 상에 암호화와 인증 기능을 포함한 애플릿을 구현하였다.



(그림 3) 자바카드 시스템 구조

3. 암호화 알고리즘

암호화 알고리즘은 키(Key)의 특성에 따라 크게 두 가지 암호화 방법으로 나뉘는데 첫째는 암복호화 하는 과정에서 암복호화 키가 같은 비밀키 암호 알고리즘(Secret-key Cipher Algorithm)이고, 둘째는 암복호화 키가 다른 비대칭 즉, 공개키 암호 알고리즘(Public-key Cipher Algorithm)이다. 이와는 달리 One-way 함수로써 데이터 값을 일정한 크기의 다이제스트(Digest)로 변환시키는 해쉬 함수 알고리즘이 있다[9]. 본 논문에서는 공개키 암호화 알고리즘인 RSA를 이용하여 암복호화를 수행하고, 해쉬 알고리즘을 이용하여 인증을 수행한다.

3.1 공개키 암호화 알고리즘(Public-key Cipher Algorithm)

대표적인 공개키 암호화 알고리즘인 RSA는 1977년 Rivest, Shamir 그리고 Adleman에 의해 제안된 방식이다. RSA는 암호화와 전자서명 모두를 제공할 수 있으며, 소인수 분해의 어려움에 안전도의 근간을 두고 있다. 즉, 두 소수 p와 q의 곱은 계산하기 쉬우나, 주어진 곱 n=pq로부터 p와 q를 추출하기는 어렵다는 사실에 근간을 두고 있다. RSA 암호화 방식은 지수 승을 가진 수식을 사용하도록 만들어졌으며, 평문은 블록으로 암호화된다. 각 블록은 어떤 수 n보다 작은 이진 값을 가진다. 암호와 복호는 평문 블록 M과 암호문 블록 C에 대하여 다음의 형태를 따른다[9].

- $C = M^e \bmod n$
- $M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$

송신자와 수신자는 n의 값을 알아야 한다. 송신자는 e의 값을 알고, 수신자만이 d의 값을 알므로 이는 공개키 {e, n}을 가진 공개키 암호 알고리즘이다. RSA 공개키 암호 알고리즘의 키 생성 과정은 다음과 같다.

1) 서로 다른 임의의 두 개의 큰 소수 p, q를 생성.
2) n = pq(단, p, q는 2^{100} 보다 큰 소수, 즉 서로 소) 와 $\varnothing = (p-1)(q-1)$를 계산.

3) gcd(e, ∅) = 1 인 정수 e(1 < e < ∅)를 임의로 선택.
4) 확장된 유클리드 알고리즘을 사용하여 ed = 1(mod ∅)인 유일한 정수 d(1 < d < ∅)를 계산.
5) 공개키 {e, n}, 개인키 {e, d}.

RSA의 암호화와 복호화는 다음과 같이 수행된다.

1) 암호화
   - 송신자는 수신자의 공개키 {e, n}을 얻는다.
   - 메시지 M을 [0, n-1]사이의 정수로 표현한다.
   - $C = M^e \bmod n$을 계산한다.
   - 암호문 C를 수신자에게 보낸다.
2) 복호화
   - 개인키 d를 이용하여, $M = C^d \bmod n$을 계산한다.

3.2 해쉬 함수(Hash Function)

해쉬 값은 다음 식과 같은 함수 H에 의해서 만들어진다.

- h = H(M)

위 식에서 M은 가변 길이의 메시지이고, H(M)는 고정 길이의 해쉬 값이다. 해쉬 값은 메시지가 정확한 것으로 판단될 경우 원 메시지에 추가된다. 수신자는 해쉬 값을 재 계산함으로써 그 메시지를 인증한다. 해쉬 함수 H는 다음의 특징들을 가져야 한다[9].

1) H는 어떤 크기의 데이터 블록에도 적용될 수 있다.
2) H는 고정된 크기의 출력을 만든다.
3) H(x)는 실질적으로 하드웨어 및 소프트웨어에 적용하기 쉬워야 하며, 어떠한 x에 대해서도 계산이 비교적 쉬워야 한다.
4) 어떠한 코드 h에 대해서도, H(x) = h인 x를 찾는 것은 계산적으로 실행 불가능하다.
5) 어떠한 블록 x에 대해서도, H(y) = H(x)인 y≠x인 것을 찾는 것이 계산적으로 실행 불가능하다.
6) H(x) = H(y)인 어떤(x, y)짝을 찾는 것이 계산적으로 실행 불가능하다. 이는 또한 강력한 충돌 회피성으로 불려진다.

주요 해쉬 함수로는 MD4, MD5, SHA-1, RIPEND-160 등이 있으며, 본 논문에서는 SHA-1 해쉬 함수를 이용하여 생성된 해쉬코드를 인증에 사용한다.

4. 시스템 설계 및 구현

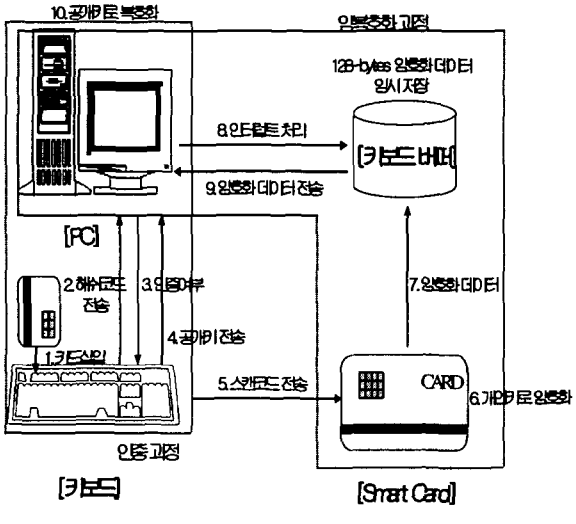
PC에서의 인증 및 입력 정보에 대한 데이터 암복호화는 현재 일부 분야에서만 이용되고 있으며, 이 경우 대부분이 PC에 인증 정보를 저장하거나 입력 구동프로그램과 연계된 암복호화 프로그램을 저장하여 놓고 수행하고 있다. 이와 같은 방법은 스마트카드를 이용하는 본 논문의 방식과 비교하여 볼 때, 구동 상의 편리함은 일부 있으나 이동성과 보안성 면에서 불리함을 알 수 있다.

4.1 시스템 구조 및 흐름도

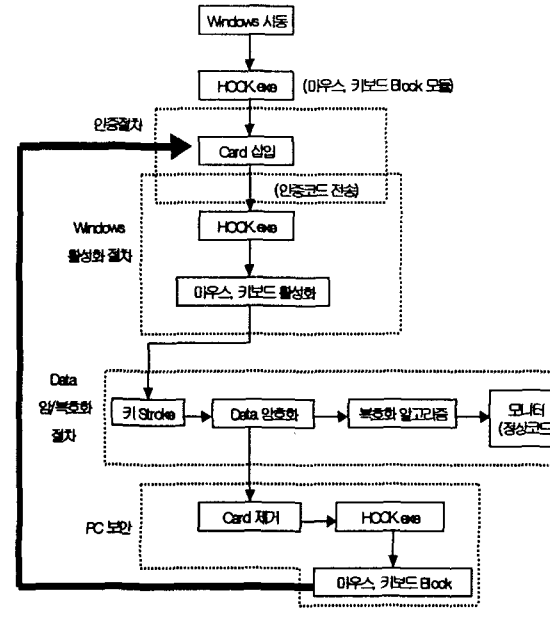
본 논문에서 제안한 스마트카드 기반의 효율적인 해킹 방지 시스템에 대한 간략한 시스템 구조 및 시나리오는 그림 4와 같다. 시스템은 크게 두 부분으로 구성할 수 있다. 첫 번째는 PC사용 권한을 얻기 위한 인증 과정이고, 두 번째는 키보드로부터 입력되는 스캔코드(Scan

Code)에 대한 암복호화 과정이다. 장차 유비쿼터스 환경에서는 두 번째 기능이 유익하게 사용될 수 있으리라 예상된다.

그림 4에서 카드를 키보드에 삽입하는 순간 인증을 위한 해쉬코드가 PC로 전송되고, PC에 이미 저장되어 있는 해쉬코드와 비교 검증되어 인증여부가 판별된다. 인증이 성공되면 데이터 복호화에 필요한 공개키를 카드에서 PC로 전송하여 저장한다. 이후 키보드로부터 발생되는 스캔코드는 카드에서 개인키에 의해 암호화되어 키보드 버퍼에 임시 저장된다. 이렇게 임시 저장된 암호화 데이터는 PC에서의 인터럽트처리에 의해 PC로 전송되어 사전에 전송받은 공개키를 이용해 복호화 된다. 특히 키의 안전한 관리를 위해 개인키와 공개키를 모두 카드에 저장하고 있으며 인증이 성공된 후에 공개키만을 PC로 전송하여 복호화에 이용하도록 한다. 또한 카드를 키보드로부터 제거하게 되면 PC에 저장되어 있던 공개키는 자동으로 삭제되어 키 관리에 대한 안전한 운용성을 갖게 된다.



(그림 4) 시스템 구조 및 시나리오



(그림 5) 시스템 흐름도

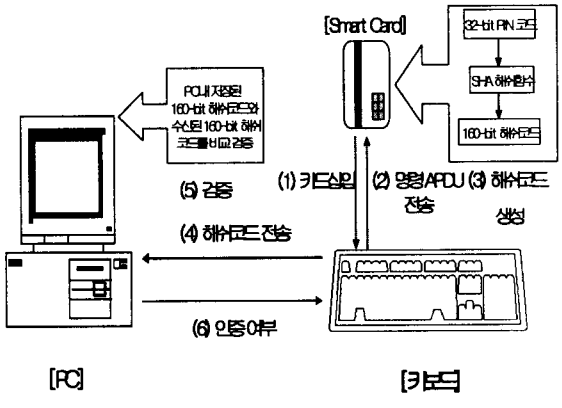
그림 5는 개괄적인 시스템 흐름도로서 카드의 삽입과 제거에 따른 키보드와 마우스의 동작과 인증 및 암복호화 절차를 보여주고 있다. 즉, 카드가 제거된 상태에서는 다시 카드삽입 단계를 거쳐야만 시스템 기능을 정상적으로 수행할 수 있다.

4.2 PC사용자 인증 기능

PC사용자 인증은 접근 권한을 가진 사용자만이 적법한 사용 권한을 얻기 위한 절차이다. 최초 Windows가 구동되면 카드의 삽입과 PIN코드에 대한 인증 없이는 키보드와 마우스가 동작 불가 상태에 있어 PC를 사용할 수 없으며, 카드의 삽입과 동시에 PIN코드 인증이 성공적으로 완료되면 키보드와 마우스의 동작이 가능하게 된다. PC사용자가 작업 중 자리이탈 시 카드를 제거하게 되면, 키보드와 마우스는 다시 동작 불가 상태로 전환된다. 본 시스템에서는 인증을 위해 SHA-1 해쉬 함수를 사용하

며 PC사용자 인증은 다음과 같은 절차에 의해 수행된다.

1) 카드를 삽입한다.
2) PIN(Personal Identification Number)코드로 사용될 카드의 유일한 Serial Number를 SHA-1 해쉬 함수를 통해 해쉬코드로 변환한다.
3) 생성된 해쉬코드를 PC로 전송한다.
4) 수신된 해쉬코드를 PC내에 저장된 해당 해쉬코드와 비교 검증한다.

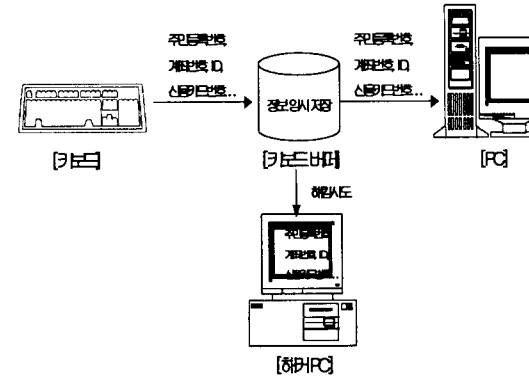


(그림 6) 인증 처리과정

그림 6은 PC사용자 인증 처리과정을 나타낸 것이다. 인증 처리과정에서 PIN코드로 사용될 카드의 유일한 Serial Number를 SHA-1 해쉬 함수를 통해 160-bit의 해쉬코드로 생성한다. 해쉬 함수의 특성상 생성된 해쉬 코드는 인증과 무결성을 제공하며, 또한 일방향성을 갖는 해쉬 함수의 특성으로 인해 제3자에 의해 해쉬코드가 노출이 되어도 역으로 PIN코드를 해독할 수 없다. 생성된 160-bit 해쉬코드는 PC로 전송이 되며, PC에서는 사전에 저장되어 있는 PIN코드에 대한 해쉬코드와 카드로부터 수신된 해쉬코드를 비교 검증한다.

4.3 키보드 버퍼 해킹 방지 기능

키보드 버퍼 해킹(Keyboard-buffer Hacking)을 방지하기 위한 목적으로, 이 과정은 시스템에서 가장 중요한 역할을 한다. 그림 7은 키보드 버퍼 해킹과정을 보여준다. 키보드 버퍼 해킹은 키보드의 키를 눌렀을 경우 키보드 버퍼로 해당키의 정보가 전송되어 임시 저장된 후 PC에서 인터럽트 처리를 통해 저장된 정보를 가져와 처리를 하는 과정에서 키보드 버퍼에 저장된 정보가 해킹시도에 의해 노출되는 것을 말한다.



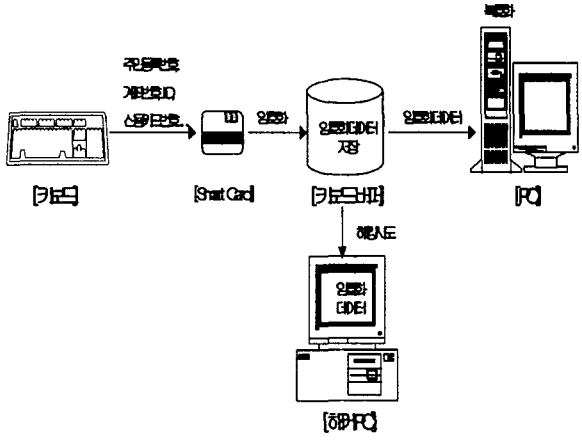
(그림 7) 키보드 버퍼 해킹과정

키보드 버퍼 해킹 방지는 키보드 버퍼에 정보가 저장되기 전 단계에서 암호화 수행을 함으로서 이루어진다. 키보드를 통해 입력되는 모든 정보를 실시간으로 암호화함으로써 제3자가 정보를 빼낸다 해도 그 정보는 의미를 상실하게 된다. 따라서 본 논문에서는 키보드 버퍼 해킹 방지를 위한 암복호화 기능을 설계하였고, 그에 따른 시스템 구조는 그림 8에 나타나며 전체 데이터 암복호화 수행 절차는 다음과 같다.

1) PC는 인증 성공 후 카드로부터 공개키를 전송 받는다.
2) 키보드로부터 카드로 스캔코드(Scan

Code)가 입력된다. 이 스캔코드는 문자의 논리적인 순서와 무관하며 오직 키보드 버튼이 실제 나열된 순서에 의해 생성된다.

3) 카드 내에 구현되어 있는 공개키 암호화 기법을 이용해 스캔코드를 개인키로 암호화한다.
4) 암호화된 데이터를 키보드 버퍼로 전송한다.
5) PC에서 인터럽트 처리에 의해 키보드 버퍼로부터 암호화 정보를 가져온다.
6) 공개키로 복호화 한다.



(그림 8) 키보드 버퍼 해킹 방지 시스템

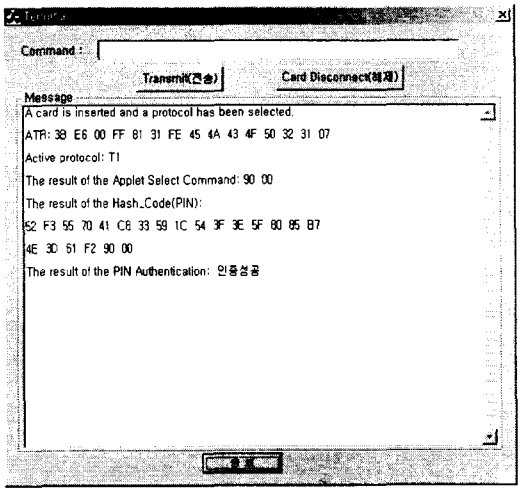
5. 시스템 성능 평가

본 장에서는 스마트카드를 기반으로 한 PC 인증 및 키보드 버퍼 해킹 방지 시스템 구현에 대한 성능을 평가 한다. 카드에서 수행될 Applet은 JCOP Tools 2.2를 사용하여 개발하였다. JCOP Tools 2.2는 자바카드 Applet을 구현하기 위한 Tool로써, 실제 카드에서 수행되는 것과 동일한 응용프로그램 구현이 가능하다[10]. 또한 Applet을 탑재할 카드로는 IBM에서 제공한 JCOP-21id 자바카드를 사용하였고, DES와 RSA 암호화 알고리즘 그리고 SHA / MD5 등의 해쉬 함수를 지원한다[10]. 시스템 개발 환경은 다음 표 1과 같다.

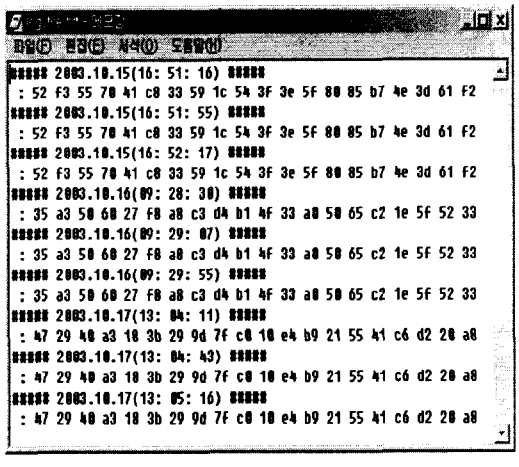
(표 1) 시스템 개발환경

운영체제	Windows 2000 Advance Server	
하드웨어	CPU	Pentium Ⅳ 1.7 GHz
	RAM	256 MB
Java Card Spec.	ROM: 64KB, RAM: 2300Bytes, EEPROM: 16KB, clock rate: 3.57MHz	
개발도구	JDK 1.3	
	JCOP Tools 2.2 (IBM), VC++	
	JCOP 21id Card (Java Card)	
	CHIPDRIVE micro100 v4.30(IFD)	

그림 9는 PC인증에 대한 결과이다. 카드를 삽입하면, PC에서는 인증을 위한 해쉬코드 생성 명령을 카드로 보낸다. 이에 따라 카드에서는 PIN코드로 사용되는 카드의 Serial Number로 160-bit 해쉬코드를 생성하여 PC로 전송하며, PC에 저장되어 있는 해당 160-bit 해쉬코드와 비교 검증하여 인증을 수행한다. 여기서 일방향성을 갖는 해쉬함수를 사용함으로서 제3자에 의해 해쉬코드가 노출되어도 역으로 PIN코드를 해독할 수 없어 PIN의 안정성을 보장받게 된다. 만약 제3자에 의한 PC접근 시도가 있을 경우 인증은 실패되며, 제3자의 PIN코드에 대한 해쉬코드와 날짜, 시간 등이 로그파일(log-file)에 기록된다. 이 로그파일은 차후 불법 접근시도 여부를 확인하기 위한 조사 자료로 활용될 수 있다. 그림 10은 인증이 실패되어 기록된 제3자의 PIN코드에 대한 해쉬코드와 날짜, 시간이 기록된 로그파일을 보여준다.

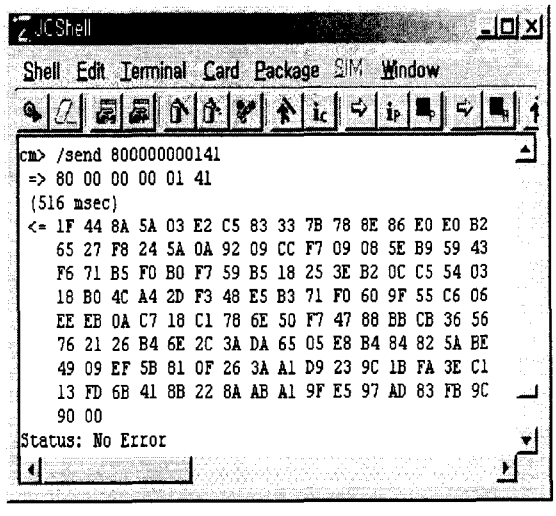


(그림 9) PC인증 결과



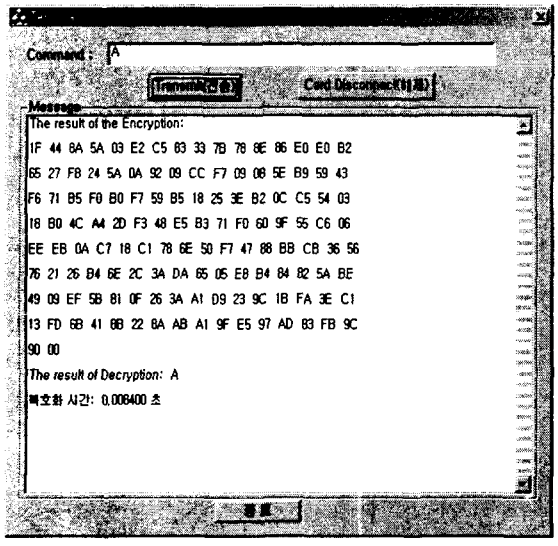
(그림 10) 인증되지 않은 해쉬코드 로그파일

인증이 완료되면 PC는 복호화를 위한 공개키를 카드로부터 전송받게 된다. 키의 노출방지를 위해 카드에 개인키와 공개키를 모두 저장하고, 인증이 성공할 경우에만 공개키를 전송하며 카드 제거 시에는 다시 PC에서 공개키가 삭제되므로 키의 안전성을 보장 받는다. 인증 완료 이후 키보드와 마우스의 동작이 가능하게 되어 PC사용이 가능하다. 인증 확인 후부터는 키보드로부터 입력되는 정보에 대하여 암복호화를 수행한다. 이 과정에서 PC 사용자가 자리이탈 시 카드를 제거하게 되면 키보드와 마우스는 동작 불가 상태가 되고 화면보호기(Screen Saver)가 실행되어 작업내용을 볼 수 없게 된다. 이러한 기능은 모두 키보드와 마우스 후킹 처리를 통해 컨트롤된다. 시스템은 PC사용의 유연성을 위해 카드 제거 시 키보드와 마우스의 동작만을 불가능 하게함으로써, PC의 종료 혹은 재 부팅 없이 카드 재 삽입을 통해 PC 작업을 계속할 수 있다.

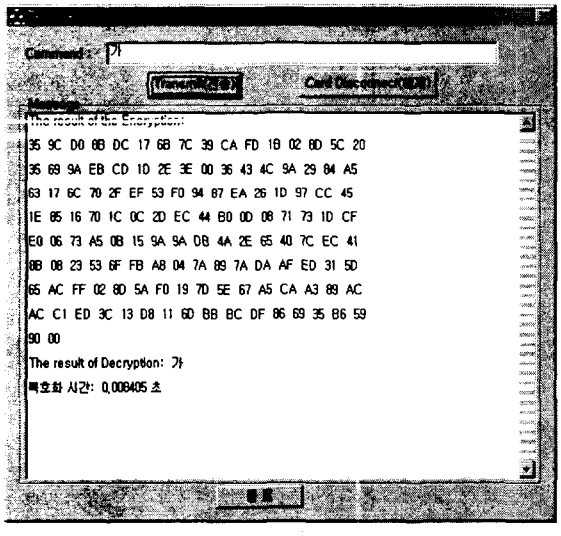


(그림 11) 개인키를 이용한 암호화 결과 및 소요 시간

그림 11은 JCOP Tools 2.2를 통해 실제 카드에 탑재된 Applet이 데이터를 개인키로 암호화 한 결과와 그에 따른 암호화 시간이다. 키보드를 통해 'A'라는 키 값의 스캔코드가 전송되면 개인키를 이용해 암호화를 수행하여 128-Byte의 암호화 데이터를 생성한다. 생성된 암호화 데이터는 키보드 버퍼로 전송되고, 임시 저장 된 후 PC에서의 인터럽트 처리에 의해 데이터를 처리한다. 따라서 버퍼의 정보를 해킹 하더라도 무의미함을 알 수 있다. 이 때 암호화에 걸린 시간이 516 msec임을 보여준다.



(a)알파벳에 대한 복호화 결과 및 소요 시간



(b) 한글에 대한 복호화 결과 및 소요 시간

(그림 12) 공개키를 이용한 복호화 결과 및 소요 시간

또한 그림 12는 PC에서 수신한 128-Byte의 암호화 데이터를 공개키로 복호화한 결과와 소요 시간이 대략 0.0084초임을 보여준다. 따라서 이는 1초에 두 타 정도의 타이핑 속도를 나타내며, 빠른 속도의 30% 정도임을 알 수 있다. 본 시스템에서는 Clock Rate 3.57MHz, 64KB ROM, 2300Bytes RAM, 16KB EEPROM의 성능을 갖는 자바카드를 이용해 시뮬레이션을 하였다. 그러므로 실제 암복호화 없이 키보드로부터 입력된 정보가 화면에 디스플레이 되는 시간과는 다소 차이가 있음을 확인 할 수 있다. 그러나 이 Delay 시간은 향후 상업성을 갖는 스마트카드(Clock Rate: 10~20MHz, ROM: 128KB, RAM: 4KB, EEPROM: 64KB)에서는 카드에서의 처리 속도가 CPU의 작동 속도에 비례하여 향상되므로 전혀 문제가 되지 않는다.

6. 결 론

본 논문에서는 개인정보 노출 방지 시스템과 효율적인 PC인증 시스템을 구현하였다. PC인증 기능을 위해 SHA-1 해쉬 함수를 사용하여 접근 권한을 가진 PC사용자에 대한 인증을 수행하였다. 여기에 최근 급격한 발전과 사용 증가 추세를 보이고 있는 스마트카드를 이용함으로서, 보다 안전하고 효율적인 보안 시스템으로서의 구체적인 방향을 제시하였다. 또한 간편하고 안전한 인증을 위해 이에 사용할 PIN 코드로 카드의 Serial Number를 사용하였고, 더욱이 PIN코드의 안정성 확보를 위해 일 방향 해쉬함수를 사용하였다. 한편 불법적인 PC 접근 시도 시 이에 대한 정보를 로그파일에 기록하여 차후 불법 접근시도 여부를 확인하는 정보로도 이용 할 수 있게 하였다. 키의 안전한 사용을 위해 카드에 개인키와 공개키 모두를 저장하여 인증이 성공할 경우에만 공개키를 PC로 전송하고, 카드 제거 시에는 다시 공개키가 PC에서 삭제되도록 하였다. 그리고 PC사용의 유연성을 위해 키보드와 마우스를 컨트롤하는 기능을 구현하여 PC작업 중 자리이탈 시 카드를 제거하게 되면, PC의 종료나 재 부팅 없이 키보드와 마우스만을 동작 불가시킴으로

서 작업 중인 데이터를 그대로 보존할 수 있다. 또 제3자를 통한 작업 데이터 노출방지를 위해 카드제거 시 PC화면에 화면보호기(Screen Saver)를 실행시켜 데이터를 보호할 수 있는 기능을 구현하였다.

키보드 버퍼 해킹 방지를 위한 방안으로, 키보드로부터 입력되는 스캔코드를 암호화하여 키보드 버퍼에 저장함으로써 해킹 혹은 제3자로부터의 정보 노출에 대한 해결책을 마련하였다. 특히, 장차 다가오는 유비쿼터스 컴퓨팅 환경에서 임의의 단말기 사용 시, 개인 입력 정보의 보호를 위해 본 논문에서 제안한 방법으로 키보드 버퍼 해킹 방지 기능을 효율적으로 구현할 수 있다. 또한, 데이터 암복호화를 위해서 공개키 암호화 알고리즘을 사용함으로써 본 시스템뿐만 아니라, 다기능 스마트카드로서 금융, 행정, 전자상거래 등 여러 다른 응용분야로의 확장이 가능하다. 키보드를 통해 입력되는 정보의 암복호화 처리 성능은 타이핑된 정보가 PC화면에 디스플레이 되는데 소요되는 시간으로 측정 된다. 본 시스템은 저성능의 카드를 이용해 시뮬레이션을 하였고, 따라서 키보드를 통해 입력되는 정보를 암복호화 없이 PC화면에 디스플레이 하는 시간보다 다소 Delay가 있음을 확인하였다. 그러나 향후 상업성을 갖는 스마트카드(Clock Rate: 10~20MHz, ROM: 128KB, RAM: 4KB, EEPROM: 64KB)에서 구현 시 카드의 처리 속도가 CPU의 작동 속도에 비례하여 향상되므로 전혀 문제가 되지 않는다.

본 논문에서 구현한 스마트카드 기반의 효율적인 PC인증과 키보드 버퍼 해킹 방지 시스템은 차후에 한층 더 효율적인 ECC 알고리즘(Elliptic Curve Cryptography Algorithm)이 지원되는 스마트카드의 활성화 시 그대로 적용됨으로서, 더욱 빠르고 보안능력이 향상된 성능을 갖추리라 기대된다.

참 고 문 헌

[1] 김성준, 이주영, 이재광, "자바카드 기반 RSA 알고리즘 구현", 한국정보처리학회 추계 학술 발표 논문집, Vol.8, No.2, pp. 839-842, 2001.

[2] 김증섭, 조병호, 김효철, 이종국, 유기영, "다양한 응용을 위한 스마트카드 운영체제", 정보과학회지 논문지, 제8권 제3호, pp.277-288, 2002.

[3] ISO/IEC 7816-1, Identification cards-Integrated circuit(s) cards with contact-Part 1: Physical characteristics, 1998.

[4] ISO/IEC 7816-2, Identification cards-Integrated circuit(s) cards with contact-Part 2: Dimensions and location of the contacts, 1999.

[5] ISO/IEC 7816-3, Identification cards-Integrated circuit(s) cards with contact-Part 3: Electronic signals and transmission protocols, 1997.

[6] ISO/IEC 786-4, Identification cards-Integrated circuit(s) cards with contact-Part 4: Interindustry commands for interchange, 1995.

[7] "Java Card 2.0 Programming Concepts", Sun MicroSystem, Inc. 1997.

[8] Zhiqun Chen, "Java CardTM Technology for Smart Cards", Sun MicroSystem.

[9] 최용락, 소우영, 이재광, 이임영, "컴퓨터 통신 보안", 도서출판 그린, 2001.

[10] http://www.zurich.ibm.com/jcop/products/cards.html

■ 저자소개

황 선 태(Su ntae Hwang)

1979 서강대학교 수학과 학사
1979~1982 KIST 연구원
1987 Case Western Reserve University(미국) 전자계산학과 석사
1993 Case Western Reserve University(미국) 전자계산학과 박사
1993~1995 현대전자연구소 책임연구원
1995~현재 대전대학교 정보통신공학과 부교수
관심분야 : 정보보안, 스마트카드 기술/응용
E-mail : hwang@dju.ac.kr

박 종 선(Jong-Sun Park)

2002 대전대학교 정보통신공학과 학사
2003 KISTI Internship 수행
2004 대전대학교 정보통신공학과 석사
2004~현재 대전대학교 한방병원 전산실 근무
관심분야 : 정보보안, 스마트카드 응용
E-mail : jspark05@hanmail.net

◈ 이 논문은 2004년 2월 15일 접수하여 1차 수정을 거쳐 2004년 5월 31일 게재확정되었습니다.