자바카드를 이용한 키보드 해킹 방지 시스템 구현

박종선, 황선태

Implementation of Keyboard-hacking Prevention System using a Java Card

Jong-Sun Park, Suntae Hwang

Abstract In this paper, we describe the system which can protect the major information from keyboard-hacking. These days the securing information matters for pc-users are becoming more important as the internet business grows rapidly, and the ubiquitous computing environment is open for everyone. Therefore, the keyboard-inputting information is necessary to be protected properly against the malicious attack.

In this paper, we propose the keyboard-hacking prevention system using a Java Card, and show the conveniency and efficiency in the results.

1. 서론

인터넷의 활성화에 따라 스마트카드(Smart Card)의 사용이 전 세계적으로 급증하고 있으며, 이와 관련한 기술 개발이 활발히 이루어지고 있다[김성준 2001]. 특히 앞으로의 정보통신분야는 유비쿼터스 컴퓨팅(Ubiquitous Computing)환경으로 바뀌어 가고 있는 추세이다. 이와 관련된 기술과 제품이 점차 확대되어 가고 있으며, 이에 맞는 보안 시스템 환경이 구축되고 개발되어야 한다. 스마트카드를 이용한 개인의 보안 환경 제공은 이러한 유비쿼터스 컴퓨팅환경에서 안전하게 시스템을 이용할 수 있게 해준다.

모든 인터넷 기반 환경에서 개인의 정보는 바이러스(Virus)와 해킹(Hacking)에 의해 항상 노출 될 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 키보드 해킹(Keyboard-hacking)으로 인해 많은 개인 정보가 노출되고 있다.

키보드 해킹은 키보드를 통해 PC로 정보가 입력되는 과정에서 키보드 버퍼로 임시 저장되는 ID, 패스워드, 신용카드번호 등을 빼 내가는 것을 말한다. 이러한 경우 PC사용자는 해킹으로부터 완전히 노출된 상태가 되며, PC사용자는 이것을 보호할 방법이 없게 된다. 이러한 문제를 해결하기 위해 현재 많은 보안 메커니즘이 연구되고 있으며, 특히 공개키 기반

암호화 기법(Public Cryptography)을 이용한 시스템들이 그 대표적인 것으로 사료된다.

본 논문에서는 최근 키보드 해킹과 PC사용자의 정보보호와 관련하여 키보드로부터 입력되는 모든 정보를 자바카드 플랫폼 상에서 암호화함으로써, 해킹으로부터 정보 노출을 방지하고, 유연성 있는 PC사용 시스템을 설계하고 구현하였다. 또한 이 시스템은 유비쿼터스 컴퓨팅환경에서 제삼자에 의한 개인정보 노출을 방지하기 위해 효율적이고, 용이하게 사용될 수 있다.

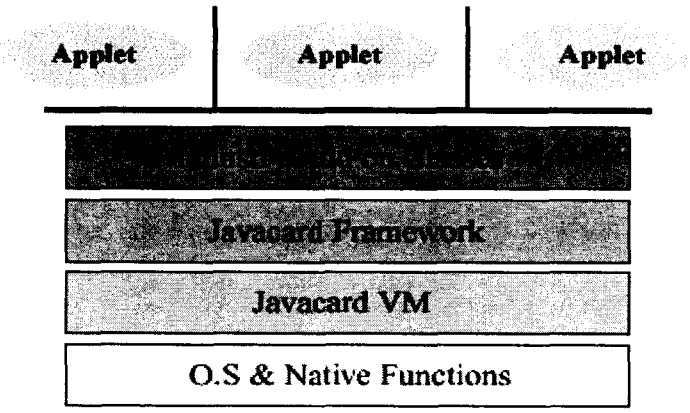
2. 자바카드

2.1. 자바카드의 정의

자바카드란 자바로 작성된 프로그램이 수행될 수 있는 스마트카드이다. 카드제조회사에 관계없이 프로그램이 수행 가능하여 플랫폼 독립적이고, 다목적 이용이 가능하며, 카드 발급 후에도 프로그램 설치가 가능하며, 객체지향 언어인 자바를 사용하므로 프로그램에 유연성이 있으며, 기존의 카드의 표준과도 호환성이 있다[김성준 2001].

2.2. 자바카드 시스템 구조

일반적으로 자바카드는 (그림 1)과 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 기술의 구현이 이루어지게 된다[Zhiqun Chen, 2000].

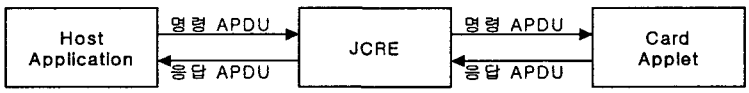


(그림 1) 자바카드 시스템 구조

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 1)에서처럼 한 장의 자바카드에는 여러 개의 어플리케이션이 존재한다. 따라서 한 장의 자바카드가 여러 가지 기능을 수행할 수 있게 함으로써 스마트카드의 활용 범위를 넓혀 줄 수 있다.

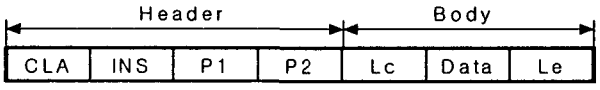
2.3. 자바카드 애플릿

자바카드 애플릿은 자바카드 상에서 실행될 수 있는 자바 프로그램이다. 자바카드 애플릿은 일반 자바 애플릿과 달리 브라우저 환경에서는 실행 될 수 없다. (그림 2)는 자바 애플릿과 호스트간의 통신을 보여준다. 애플릿과 호스트간의 통신은 명령어 APDU와 응답 APDU로 구성되는 APUD교환을 통해서 이루어진다. APDU교환은 애플릿과 호스트간에 직접 이루어지는 것이 아니라 JCRE를 매개로 하여 이루어지고, JCRE는 애플릿과 호스트간에 교환되는 APDU의 관리와 감독 역할을 수행한다[ISO/IEC 7816-4, 1995].

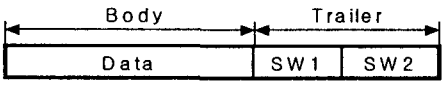


(그림 2) 애플릿 통신

APDU는 카드 상의 통신에서 사용되는 전송 메시지의 형태로 ISO 7816에 규정되어 잇다. 전송방식은 명령(Command)과 응답(Response)으로 이루어져 있다. (그림 3)는 명령과 응답 APDU의 구조를 보여준다. 명령 APDU에서 CLA는 Class byte이고, INS는 명령어 구분코드이다. P1, P2는 명령어에서 사용되는 변수 1과 변수 2이다. Lc는 송신 데이터 바이트 수를 나타내고, Le는 수신 데이터 바이트 수를 나타낸다. Data는 송/수신 데이터를 나타낸다. 응답 APDU에서 SW1, SW2는 응답 메시지에서 사용하는 상황 표시1, 상황 표시 2를 나타낸다[ISO/IEC 7816-4, 1995].



(a) 명령 APDU



(b) 응답 APDU

(그림 3) 명령 / 응답 APDU 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 알고리즘이고(Secret-key Cipher Algorithm)이고, 둘째는 암복호화 키가 다른 비대칭 즉, 공개키 암호 알고리즘(Public-Key Cipher Algorithm)이다[안형근, 2001].

3.1. 비밀키 암호화 알고리즘(Secret-Key Cipher Algorithm)

비밀키 암호 시스템의 대표적인 예는 DES이다. 비밀키 암호 시스템은 암호화키와 복호화키가 같은 시스템으로, 암호화를 할 때의 키로 복호화가 가능하다는 뜻이다. DES는 brute-force공격에 대한 잠재적인 취약성을 가지고 있다. 이에 대한 대안으로 가장 많이 쓰이는 알고리즘이 Triple-DES이다. DES는 하나의 비밀키를 사용하지만, Triple-DES는 2개의 서로 다른 키를 사용하여 암호-복호-암호의 순서로 암호화와 복호화를 수행한다. 메시지 M에 대해 암복호화 수행과정은 다음과 같다[최용락 외 3명, 2001].

- 암호화: $C = E_{k1}(D_{k2}(E_{k1}(M)))$
- 복호화: $M = D_{k1}(E_{k2}(D_{k1}(C)))$

3.2. 공개키 암호화 알고리즘(Public-Key Cipher Algorithm)

대표적인 공개키 암호화 알고리즘은 RSA이다. RSA는 1977년 Riverst, 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 방식은 지수 승을 가진 수식을 사용하도록 만들어졌다. 평문은 블록으로 암호화된다. 각 블록은 어떤 수 n보다 작은 이진 값을 가진다. 암호와 복호는 평문 블록 M과 암호문 블록 C에 대하여 다음의 형태를 따른다[최용락 외 3명, 2001].

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

송신자와 수신자는 n의 값을 알아야 한다. 송신자는 e의 값을 알고, 수신자만이 d의 값을 안다. 그러므로 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 $\Phi = (p-1)(q-1)$를 계산.
3) $\gcd(e, \Phi) = 1$ 인 정수 $e(1 < e < \Phi)$를 임의로 선택.
4) 확장된 유클리드 알고리즘을 사용하여 $ed = 1 \pmod{\Phi}$인 유일한 정수 $d(1 < d < \Phi)$을 계산.
5) 공개키 {e, n}, 비밀키 {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.

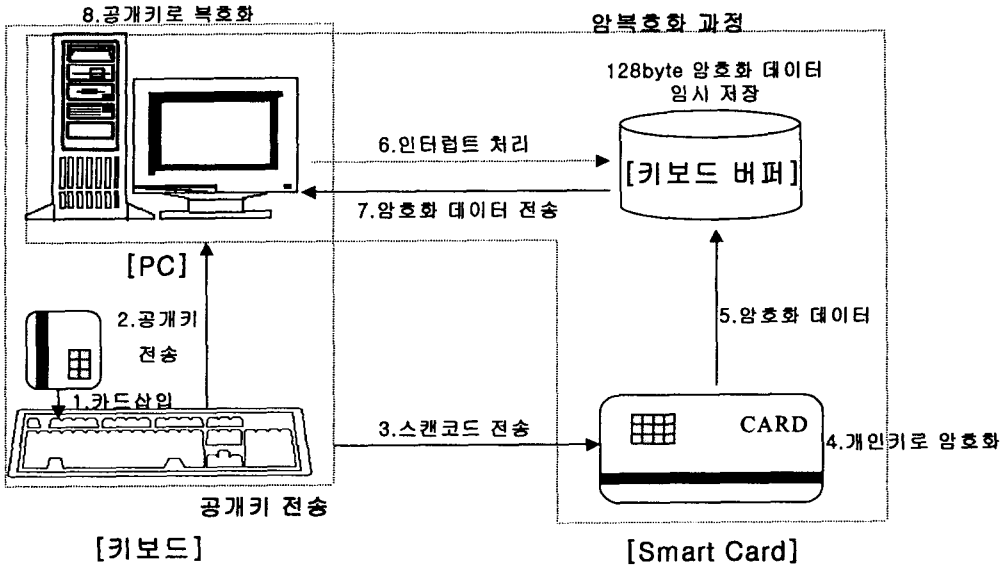
2) 복호화
- 비밀키 d를 이용하여, $M = C^d \bmod n$을 계산한다.

4. 시스템 설계 및 구현

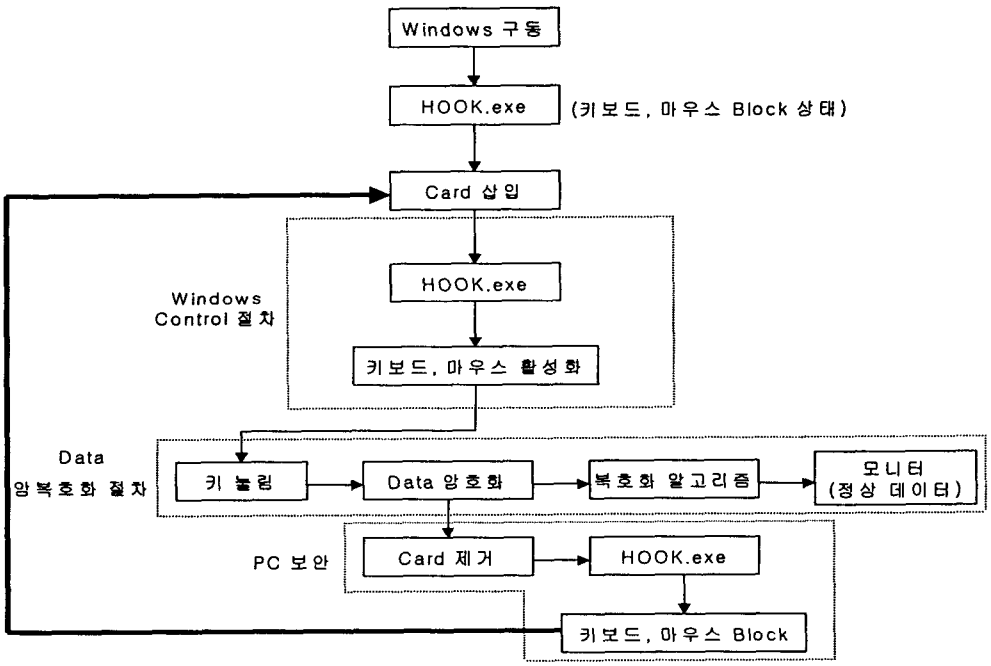
4.1. 시스템 구조 및 흐름도

본 논문에서 제안한 자바카드를 이용한 키보드 해킹 방지 시스템 구현에 대한 간략한 시스템 구조 및 시나리오는 (그림 4)과 같다.

시스템은 크게 두 부분으로 구성할 수 있다. 첫 번째는 키보드로부터 입력되는 스캔코드(Scan Code)에 대한 암복호화 절차이며, 두 번째는 카드의 삽입과 제거에 따른 Windows Control로 구성된다. (그림 4)에서 카드를 키보드에 삽입하면 키보드로부터 입력되는 정보에 대해 카드에서 개인키로 암호화된 데이터를 PC에서 복호화 하기 위한 공개키를 PC로 전송하게 된다. 이후 키보드에서 입력되는 키값에 대한 스캔코드를 카드에서 입력받아 개인키를 이용 암호화하여 암호화 데이터를 생성 후 키보드 버퍼로 전송하고, PC에서 키보드 버퍼에 임시 저장되어 있는 암호화 데이터를 가져와 사전에 전송 받은 공개키를 이용 복호화를 수행한다. 키의 안전한 관리를 위해 개인키와 공개키는 모두 카드에 저장이 되어 있으며 카드를 키보드에 삽입한 후에 공개키만을 PC로 전송하여 암복호화에 이용한다. 또한 카드를 키보드로부터 제거하게 되면 PC에 저장되어 있던 공개키는 자동으로 제거가 되어 키 관리에 대한 안전한 운용성을 갖게 된다.



(그림 4) 시스템 구조 및 시나리오

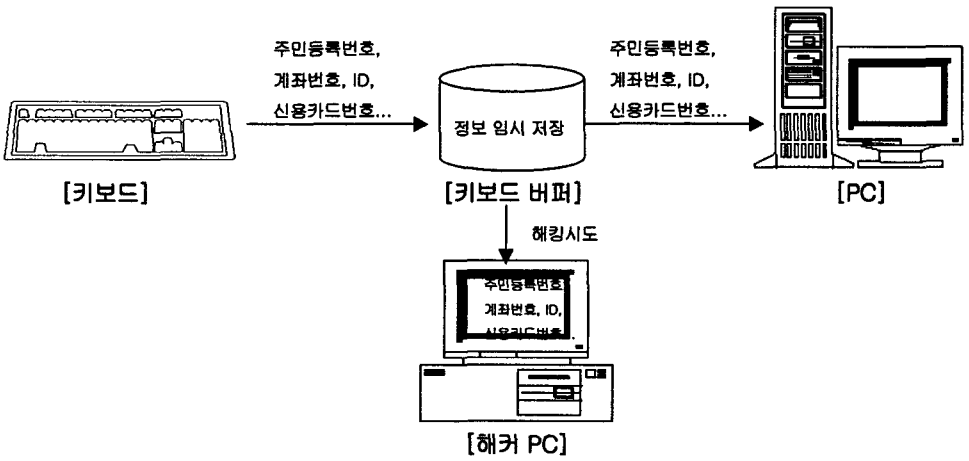


(그림 5) 시스템 흐름도

(그림 5)는 본 논문의 시스템 흐름도로서 카드의 삽입과 제거에 따른 키보드와 마우스의 기능이 유연성 있게 동작하는 것을 확인 할 수 있다. 카드가 제거된 상태에서는 다시 카드 삽입단계로 이동하여 수행하면 동일한 절차에 의한 시스템 기능이 수행된다.

4.2. 데이터 암복호화 과정

데이터 암복호화 절차는 시스템에서 가장 중요한 부분이다. 데이터 암복호화 구현의 주 목적은 키보드 해킹(Keyboard Hacking)을 방지하기 위함이다. (그림 6)은 키보드 해킹과정을 보여준다. 키보드 해킹은 키보드의 키를 눌렀을 경우 키보드 버퍼로 해당키의 정보가 전송되어 임시 저장된다. 그 후, 인터럽트 처리에 의해 키보드 버퍼에 저장된 정보를 가져와 PC에서 처리를 하게 되는데, 이 과정에서 키보드 버퍼에 저장된 정보가 해킹시도에 의해 노출되는 것을 키보드 해킹이라 한다.

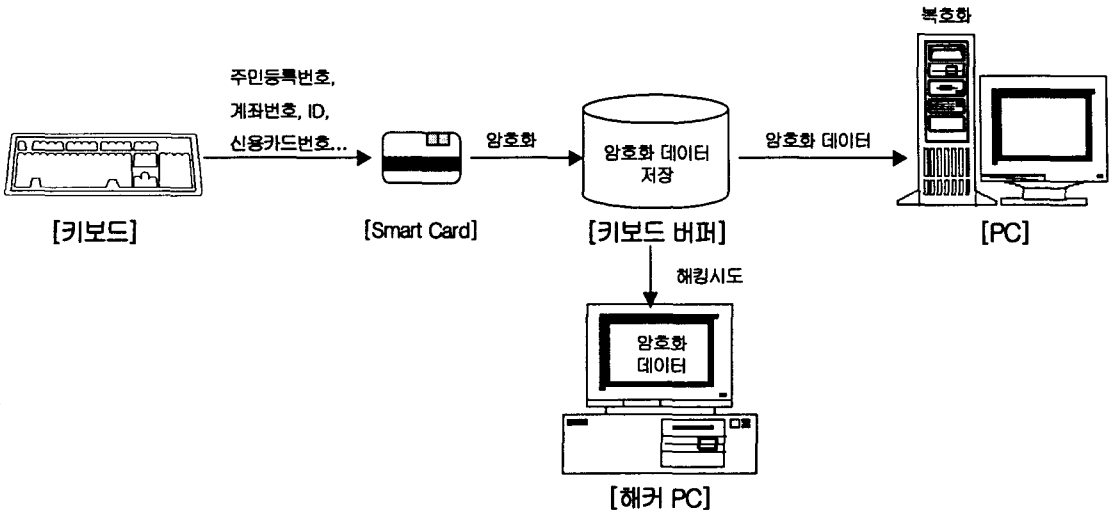


(그림 6) 키보드 해킹과정

키보드 해킹 방지는 키보드 버퍼에 정보가 저장되기 전 암호화 수행을 통해 이루어 질 수 있다. 키보드를 통해 입력되는 모든 정보를 실시간으로 암호화함으로써 제삼자가 정보를 빼낸다 해도 그 정보는 의미를 상실하게 된다. 키보드 해킹 방지를 위한 시스템 구조는 (그림 7)과 같다. 키보드에서 입력된 정보가 키보드 버퍼에 저장되는 중간 과정에 카드에서 데이터 암호화를 수행하게 된다. 전체 데이터 암복호화 수행 절차는 다음과 같다.

1) 키보드로부터 카드로 스캔코드(Scan Code)를 입력받는다.
2) 카드 내에 구현되어 있는 공개키 암호화 기법을 이용해 개인키로 암호화한다.
3) 암호화된 데이터를 키보드 버퍼로 전송한다.
4) PC에서 인터럽트 처리에 의해 키보드 버퍼로부터 데이터를 가져온다.
5) 공개키 암호화 기법의 공개키로 복호화 한다.

스캔코드는 문자의 논리적인 순서와 무관하며 오직 키보드 버튼이 실제 나열된 순서에 의해 생성된다. 본 논문의 시스템은 이 스캔코드 정보에 대해 암복호화를 수행한다.



(그림 7) 키보드 해킹 방지 시스템

카드에 구현된 Applet을 통해 키보드로부터 입력되는 정보를 개인키로 암호화 처리하고, PC에서 암호화된 정보를 수신하여 공개키로 복호화 하게 된다. 이 과정에서 암호화 데이터는 키보드 버퍼에 임시 저장이 되는데 해킹을 통해 정보가 누출이 되어도, 그 정보는 의미를 상실하게 된다. 따라서 이러한 시스템을 통해 키보드 해킹을 효율적으로 방지 할 수 있게 된다.

4.3. Windows Control 과정

Windows Control은 PC사용자의 PC사용에 대한 유연성을 갖고 제삼자로 하여금 PC접근 권한을 방지하기 위한 것이다.

최초 Windows가 구동이 되면 카드의 삽입 없이는 키보드와 마우스가 Block상태에 있어 PC를 사용할 수가 없으며, 카드의 삽입과 동시에 키보드와 마우스의 동작이 가능케 되어 PC사용이 가능하다.

PC사용자가 작업 중 짧은 시간 자리를 비울 시 카드를 제거하게 되면, 키보드와 마우스는 다시 Block상태로 전환된다. 이 과정에서 제삼자로부터 PC에 작업중인 내용 보호를 위해 카드가 제거되면 화면보호기(Screen saver)를 실행하여 작업내용을 볼 수 없게 한다. 이러한 기능은 모두 키보드와 마우스 후킹 처리를 통해 컨트롤된다. 시스템은 PC사용의 유연성을 위해 카드 제거 시 키보드와 마우스의 동작만을 Block시킴으로써, PC의 종료 혹은 재부팅 없이 카드 재 삽입을 통해 PC작업을 계속할 수 있다.

5. 시스템 구현 결과 및 분석

본 장에서는 자바카드를 이용한 키보드 해킹 방지 시스템 구현에 대한 결과와 분석을 하

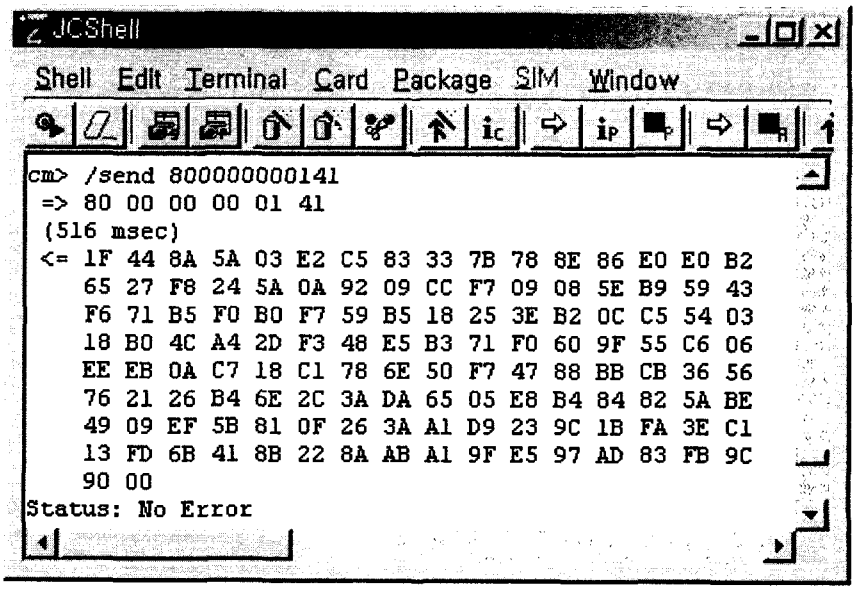
였다. 카드에서 수행될 Applet은 IBM사의 JCOP Tools 2.2를 사용하여 개발하였다. JCOP Tools 2.2는 자바카드 Applet을 구현하기 위한 Tool로써 실제 카드에서 수행되는 결과와 동일한 응용프로그램이 구현 가능하다. 또한 Applet을 탑재할 카드로는 JCOP 21id로써 IBM사에서 개발한 자바카드이며, DES, RSA 암호화 알고리즘과 SHA / MD5의 해쉬함수를 지원한다. HOOK.exe 프로그램은 주로 VC++ dll파일과 MFC를 이용하여 구현하였다.

시스템 개발 환경은 다음 <표 2>과 같다.

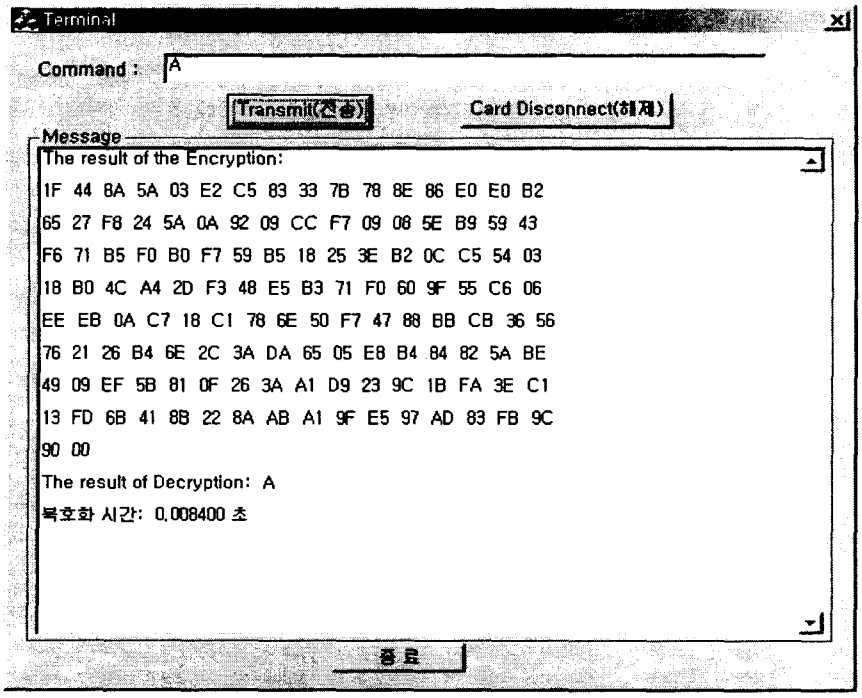
<표 2> 시스템 개발 환경

운영체제	Windows 2000 Advance Server	
하드웨어	CPU	Pentium IV 1.7 GHz
	RAM	256 MB
Card Memories	ROM: 64KB, RAM: 2300Bytes, EEPORM: 16KB, clock rate: 3.57MHz	
개발도구	JDK 1.3	
	JCOP Tools 2.2 (IBM), VC++	
	JCOP 21id Card (Java Card)	
	CHIPDRIVE micro 100 v4.30 (IFD)	

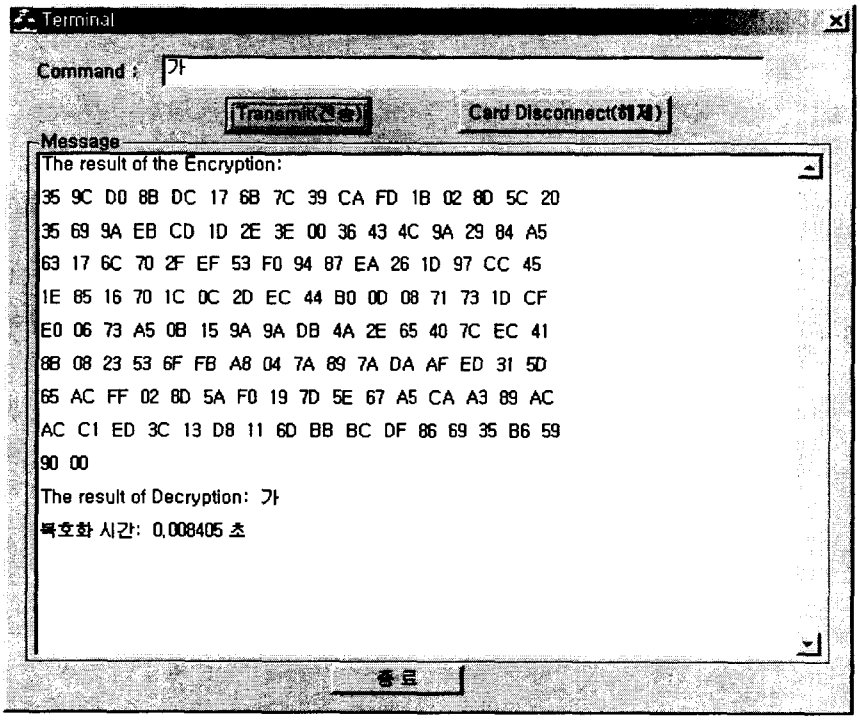
(그림 8)은 JCOP Tools 2.2를 통해 실제 카드에 탑재된 Applet이 데이터를 개인키로 암호화 한 결과와 그에 따른 암호화 시간이다. 키보드를 통해 'A'라는 키 값의 스캔코드가 카드로 전송되면 개인키를 이용해 암호화를 수행하여 128byte의 암호화 데이터를 생성한다. 이 생성된 암호화 데이터는 키보드 버퍼로 전송이 되고, 임시 저장이 된 후 PC에서의 인터럽트 처리에 의해 다시 PC로 전송된다. 이때 암호화에 걸린 시간은 516(msec) 이다.



(그림 8) 개인키를 이용한 암호화 결과 및 암호화 시간



(a) 알파벳에 대한 복호화 결과 및 복호화 시간



(b) 한글에 대한 복호화 결과 및 복호화 시간

(그림 9) 공개키를 이용한 복호화 결과 및 복호화 시간

(그림 9)는 PC에서 수신한 128byte의 암호화 데이터를 공개키로 복호화 한 결과와 복호화 시간이다. 결과적으로, 키보드로부터 입력된 정보가 카드에서 개인키로 암호화되고, PC에서 다시 공개키로 복호화 하여 정상적인 정보가 화면에 디스플레이 된다. 본 시스템은 64KB ROM, 16KB EEPROM, 2300Bytes RAM, clock rate 3.57MHz의 성능을 갖는 카드를 이용해 시뮬레이션을 하였다. 따라서, 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: 20~40MHz, ROM: 128KB, RAM: 4KB, EEPORM: 64KB)에서는 카드에서의 처리 속도가 CPU의 작동 속도에 비례하여 성능이 향상되므로 전혀 문제가 되지 않는다. PC작업 중 카드를 제거하게 되면, 시스템은 키보드와 마우스의 동작을 Block시키고, PC화면에 화면보호기(Screen Saver)를 실행시킴으로써, 제삼자로부터의 PC 데이터 보호 기능을 제공한다.

6. 결론

인터넷 사용의 증가와 활성화에 따른 정보보호 시스템의 필요가 크게 대두되고 있다. 전자상거래나 인터넷 뱅킹, 증권거래등은 열린 구조(Open Architecture)를 기반으로 이루어지고 이에 따른 정보보호 시스템의 구축은 필수 사항이 되었다. 정보보호 시스템이 발전함에 따라, 정보 노출을 위한 불법적인 행위 또한 계속해서 증가하고 있는 추세이다. 최근에는 해킹이나 바이러스에 의한 정보 노출이 빈번하였으며, 이를 방지하기 위한 시스템이 필요할 때이다.

본 논문에서는 개인 정보의 해킹 방지 시스템을 구현하였다. 특히 키보드 해킹 방지를 위한 시스템 구현을 주안점으로 두었으며, 유비쿼터스 컴퓨팅환경에 쉽게 이용 가능한 이점을 제시하였다. 또한 제삼자로부터 정보 노출 방지를 위한 시스템을 설계 구현하였다.

키보드 해킹 방지를 위한 방안으로, 키보드로 입력되는 스캔코드를 암호화하여 키보드 버퍼에 저장함으로써 해킹 혹은 제삼자로부터의 정보 노출에 대한 해결책을 마련하였다. 특히, 데이터 암복호화를 위해서 공개키 기반 알고리즘으로 대표적인 RSA를 사용하였다. DES나 Triple-DES는 암복호화 속도가 RSA보다 빠른 장점을 가지고 있지만, 안전성과 여러 응용분야에 쉽게 접근, 사용 가능하다는 면에서 RSA가 훨씬 좋은 장점을 가지고 있다. 따라서 본 논문에서 RSA를 사용함으로써 본 시스템의 응용뿐만 아니라, 다기능 스마트카드로서 금융, 행정, 전자상거래등 여러 다른 응용분야로의 활용가능성을 기대할 수 있다. 또한 본 시스템은 유비쿼터스 컴퓨팅환경에 쉽게 이용 가능한 효율적인 보안 시스템으로써의 방향을 제시하였다.

PC사용의 유연성을 위해서는 키보드와 마우스를 컨트롤하는 기능을 구현하였다. PC작업 중 자리이탈 시 카드를 제거하게 되면 PC의 종료나 재부팅 없이 키보드와 마우스만을 Block시킴으로써 작업 중인 데이터를 보존할 수 있다. 또한 제삼자를 통한 작업 데이터 노출 방지를 위해 카드제거 시 PC화면에 화면보호기(Screen Saver)를 실행시켜 데이터를 보호할 수 있는 기능을 구현하였다. 키보드를 통해 입력되는 정보를 암복호화 하는 기능은 PC화면에 타이핑된 정보가 디스플레이 되는 시간을 고려해야 한다. 본 시스템은 64KB

ROM, 16KB EEPROM, 2300Bytes RAM, clock rate 3.57MHz의 성능을 갖는 카드를 이용해 시뮬레이션을 하였다. 따라서 키보드를 통해 입력되는 정보를 암복호화 없이 PC화면에 디스플레이 하는 시간과는 다소 차이가 있음을 확인하였다. 그러나 향후 상업성을 갖는 스마트카드(clock rate: 20~40MHz, ROM: 128KB, RAM: 4KB, EEPORM: 64KB)에서는 카드에서의 처리 속도가 CPU의 작동 속도에 비례하여 성능이 향상되므로 전혀 문제가 되지 않는다.

본 논문에서 구현한 자바카드를 이용한 키보드 해킹 방지 시스템은 차후에 한층 더 효율적인 ECC(Elliptic Curve Cryptography)알고리즘이 지원되는 스마트카드의 활성화 시 그대로 적용됨으로서, 더욱 빠르고 보안능력이 향상된 성능을 갖추리라 기대된다.

참 고 문 헌

[1] 최용락, 소우영, 이재광, 이임영, 「컴퓨터 통신 보안」, 도서출판 그린, 2001.

[2] 김성준, 이주영, 이재광, "자바카드 기반 RSA 알고리즘 구현", 한국정보처리학회 추계 학술 발표논문집, Vol.8, No.2, pp839-842, 2001.

[3] 김증섭, 조병호, 김효철, 이종국, 유기영, "다양한 응용을 위한 스마트카드 운영체제", 정보 과학회지 논문 지 제8권 제3호, 2002.

[4] 이승혁, "타원곡선 암호알고리즘을 이용한 효율적인 디지털 컨텐츠 암호화 기법에 관한 연구", 석사학위 논문, 2001, 대전대학교.

[5] 안형근, "임베디드 시스템을 이용한 웹 보안 시스템 설계 및 구현 ", 석사학위 논문, 2001. 대전대학교.

[6] 이향진, 이홍섭, "공개키 기반구조와 전자서명", 전자공학회지 제29권, 제3호, 2002.

[7] 황선태, 이형, "스마트카드 모델의 기준에 관한 연구", 한국 전자거래 학회 연구논문, 제4권 2-3호, pp9-212, 1999.

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

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

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

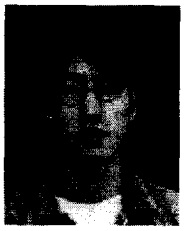
[11] ISO/IEC 7816-4, Identification cards-Inte grated circuit(s) cards with contact-Part 4: Interindustry commands for interchange, 1995.

[12] "Java CardTM 2.1.2 Development Kit User's Guide", Sun MicroSystem, Inc. 2001.

[13] "Java CardTM 2.2 Application Programming Interface", Sun MicroSystem, Inc. 2002.

[14] "Java Card Applet Developer's Guide", Sun MicroSystem, Inc. 1998.

[15] Zhiqun Chen, "Java CardTM Technology for Smart Cards", Sun MicroSystem. 2000.



박종선(Jong-Sun Pa가)
2002 대전대학교 정보통신공학과 학사
2002 ~ 현재 대전대학교 정보통신공학과 석사과정
관심분야: 정보보안, 스마트카드

황선태(Suntae Hwang)
1979 서강대학교 수학과 학사
1987 Case Western Reserve University(미국) 전자계산학과 석사
1993 Case Western Reserve University(미국) 전자계산학과 박사
1995 ~ 현재 대전대학교 정보통신공학과 부교수
관심분야: 정보보안, 스마트카드