스마트카드를 이용한 시리얼 키 인증 시스템 구현 및 설계

Design and Implementation of Serial Key Certification System Using Smartcard

김유두*, 문일영*

Yu-Doo Kim*, Il-Young Moon*

요 약

소프트웨어나 디지털콘텐츠의 저작권 보호를 위하여 시리얼 키 입력 방식의 인증 시스템이 널리 이용되고 있다. 하지만 가상의 시리얼 키를 만들거나 하나의 시리얼 키를 여러 사용자가 공유하여 저작권의 보호가 되지 않고 있다. 이에 다양한 암호화 기법을 통해 시리얼 키의 복제를 막으려는 노력이 진행되고 있지만 소프트웨어의 기술만으로는 한계가 있다. 본 논문에서는 보안성이 강화되고 휴대가 편리하여 다양한 분야에서 많이 활용되기 시작한 스마트카드를 이용하여 소프트웨어와 디지털콘텐츠의 저작권 보호를 위한 시리얼 키 인증 시스템을 제안 하였다.

Abstract

Current certification system use serial key for protect copyright of software and digital contents. But It is not efficient system, because It is not protect copyright through create virtual key or use public key. So we research on various cryptology for prevent illegal copy of serial key, but It is not enough protection of copyright that use software technology only. In this paper, we propose certification system using smart card for protect copyright of software and digital contents.

Key words : Smart card, Java card, Serial Key, Certification

I. 서 론

유형의 하드웨어와 달리 무형의 소프트웨어는 쉽게 복제가 가능하여 비용을 지불하지 않고 불법적으로 사용하는 이용자가 많다. 특히 인터넷의 급격한 발달로 인하여 누구나 쉽게 불법 복제 소프트웨어를 접할 수 있게 되었다. 또한 음반, 영상 등 디지털콘텐츠까지 무분별하게 인터넷을 통하여 불법으로 유통이 이루어지면서 저작권에 대한 사용자의 인식이 부족하여 관련 산업에 종사하는 사람들의 생존권이 위협을 받고 있다. 이에 소프트웨어 분야에서는 오래전부터 시리얼 키 방식을 통해 정품 인증을 하고 있지만, 이 또한 가상 시리얼 키 제작이나 여러 사용자가 시리얼 키를 공유하여 사용하는 등 근본적인 저작권 보호의 해결책이 되지 못하고 있다. 이를 위해 인터넷을 통한 인증과 암호화 강화 등 다양한 대책을 내놓고 있지만 소프트웨어적인 방법으로는 완벽한 저작권 보호에 한계가 있다. 그리하여 시리얼 포트에

* 한국기술교육대학교 정보미디어공학과(Dept. of Information Media Eng., Korea University of Technology and Education)
· 제1저자 (First Author) : 김유두
· 접수일자 : 2007년 11월 15일

연결하는 하드웨어를 이용한 인증방법이 나오기도 하였지만 소프트웨어의 추가에 따라 계속 하드웨어를 추가하는 것은 불편하고 현실성이 없어 좋은 대안이 되지 못하였다.

이에 본 논문에서는 보안성이 강하고 소프트웨어의 추가에 따른 별도의 하드웨어의 추가 장착이 필요 없는 스마트카드를 활용한 시리얼 키 인증 시스템을 제안하였다.

Ⅱ. 스마트카드 기술

스마트카드는 집적 회로 기억장치와 중앙처리장치를 탑재한 반도체 칩이 내장되어 있는 카드로 자기 카드에 비해 기억용량이 크고 처리 능력이 있어 똑똑한 카드라는 의미로 불리고 있다[1]. 특히 보안성이 뛰어나 은행 카드, 신분 증명 카드 등에 널리 활용 되어 지고 있다.

2-1 스마트카드

스마트카드는 작은 컴퓨터와 같아서 8, 16, 32비트 마이크로프로세서, EEPROM(Electrically Erasable Programmable Read-Only Memory), RAM(Random Access Memory), ROM(Read Only Memory), RSA(RonRivest, Adi Shamir, and Leonard Adleman), DES(Data Encryption Standard), AES(Advanced Encryption Standard)등과 같은 보안 관련 프로세서, 그리고 작지만 운영체제까지 탑재하고 있다[1][2].

스마트카드는 단말기와 카드리더를 통해 서로 정보를 주고받기 위한 방법으로 일반적으로 접촉식(Contact)과 비접촉식(Contractless)으로 구분한다[1].

그림 1은 접촉식 스마트카드를 보여주고 있는 것으로, 카드를 수용하는 인터페이스장치(IFD : Interface device)에 삽입 되었을 때 카드의 IFD 접점에 접촉 됨으로써 전원이 공급되어 카드가 활성화 되는 상태이다.

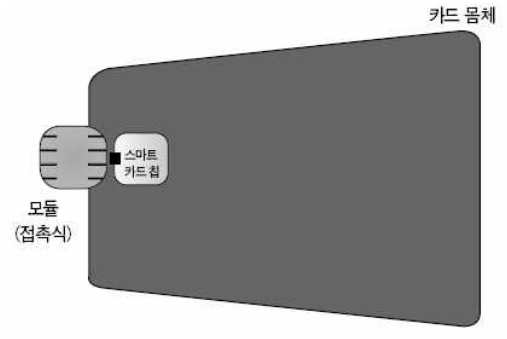

그림 1. 접촉식 스마트카드 내부 구성도
Fig. 1. Structure of Contact Smartcard

그림 2는 비접촉식 스마트카드로, 정보처리 기능에 필요한 연산소자와 기억소자는 접촉식 카드와 동일하지만 카드 내의 칩을 구동하기 위한 전원공급이 카드 내의 안테나를 통해 유도 전류를 발생해 이뤄지는 형태의 카드이다.

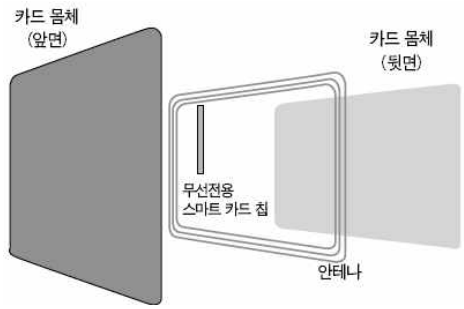

그림 2. 비접촉식 스마트카드 내부 구성도
Fig. 2. Structure of Contactless Smartcard

그 외에도 접촉식 카드와 비접촉식 카드를 물리적으로 한 개의 칩으로 통합한 형태의 콤비카드가 있으며, 접촉식 카드와 비접촉식 카드가 물리적으로 분리된 상태로 카드에 탑재된 하이브리드 스마트카드가 있다.

그림 3. 다양한 형태의 스마트카드
Fig. 3. Various Smartcard

2-2 자바카드

자바카드는 응용서비스 개발업체가 독립적인 응용서비스를 개발하여 이미 구현된 시스템에 쉽게 추가하여 기능을 확장할 수 있도록 Sun Microsystems에서 개발한 자바 언어 스펙 및 이를 구현한 다기능 스마트카드를 말한다[3].

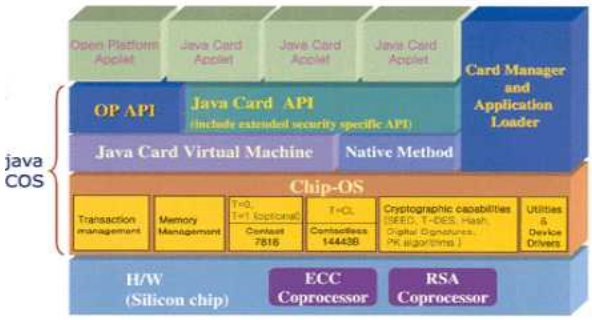

그림 4. 자바카드 OS의 기능적인 구성
Fig. 4. Structure of JavaCard OS

자바카드는 기본 자바 언어에 비하여 언어적인 제한이 많이 있다. 아래의 표 1 에서는 이러한 자바카드의 언어적 제약을 설명하고 있다[3].

표 1. 자바카드의 언어적 제약
Table 1. Restriction of JavaCard

지원되는 기능	지원되지 않는 기능
• 작은 크기의 데이터 : Boolean, byte, short • 1차원 배열 • 자바 패키지, 클래스, 인터페이스, 예외처리 • 자바객체 지향 특성 : 상속성, 가상 Method, 오버로딩 등	• 큰 크기의 데이터 타입 : long, double, float • 문자와 문자열 • 다차원 배열 • 동적 클래스 로딩 • 보안 관리자 • 메모리 가비지 컬랙션, 쓰레드 • 오브젝트 클로닝 • 오브젝트 Serialization

지속적인 개발 Toolkit의 발전에 따라 위의 제약 기능들을 개선하고 있기 때문에 항상 최신 버전의 사용을 염두하고 개발을 하여야 한다.

2-3 APDU

자바카드는 기존 스마트카드 표준을 지원하고 수용한다. 따라서 스마트카드의 통신을 위한 언어는 ISO 7816에 정의된 APDU라는 것을 이용한다[1].

자바카드 애플릿은 호스트 애플리케이션과 통신할 때 수동적인 입장에 항상 있으므로 호스트 애플리케이션이 주는 명령(Command APDU)를 기준하여 동작하고 이에 반응하여 응답(Response APDU)을 전송한다[4].

표 2. 명령 APDU
Table 2. Command APDU

Header (필수)				Body (선택)		
CLA	INS	P1	P2	Lc	Data	Le

표 2 는 명령 APDU의 구조를 나타낸다. 기본적으로 Header와 Body로 구분하는데 반드시 Header에 명령어와 해당 명령어의 입력 값을 정의한다. 또한 각 값들은 16진수로 구성되어 있다.

표 3. 응답 APDU
Table 3. Response APDU

Body (선택)	Status Word (필수)	
Data	SW1	SW2

표 3 은 응답 APDU를 보여주고 있다. 스마트카드에 명령이 전달되면 해당 명령어에 대한 데이터를 호스트 애플리케이션에게 응답하는 것이다. 아래의 그림 5 는 응답 코드에 대한 구조를 보여준다.

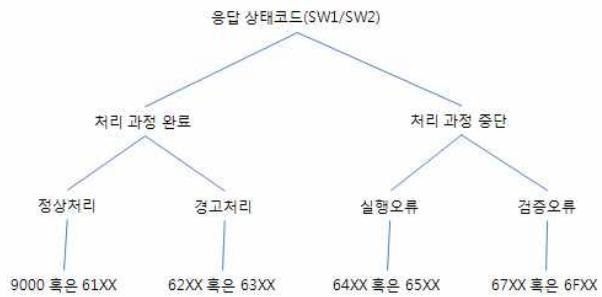

그림 5. 응답 상태 코드 구조도
Fig. 5. Structure of Response Status Code

Ⅲ. 시스템 설계

본 논문에서 구현하는 시스템은 스마트카드 에서의 처리를 위한 스마트카드 애플릿, PC에서 시리얼 키 인증을 위한 MFC(Microsoft Foundation Class)기반의 애플리케이션으로 구성되어 있다.

3-1 하드웨어 구성

그림 6은 시스템의 하드웨어 구성을 보여주고 있다. 하드웨어는 스마트카드와 스마트카드의 데이터 통신을 위한 리더, 그리고 시리얼 키 인증을 요청하는 애플리케이션이 설치된 PC로 구성되어 있다.

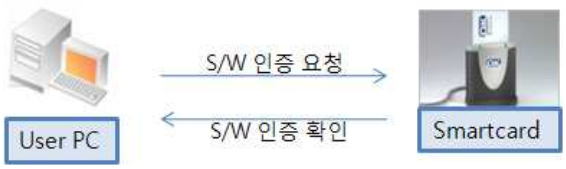

그림 6. 하드웨어 구조
Fig. 6. Hardware structure

기본 동작은 사용자의 PC에서 저작권이 있는 소프트웨어 또는 디지털콘텐츠를 실행하면 시리얼 키 요청 메시지를 스마트카드 리더기에 보내고 리더기는 스마트카드를 동작시켜 요청한 시리얼 키를 PC에 전송하거나 인증 실패 메시지를 전송한다.

3-2 통신 프로토콜

스마트카드와 PC의 통신을 위해서는 프로토콜을 설계하여야 하는데 본 논문에서 구현한 시스템은 시리얼 키 등록, 인증, 삭제의 세 가지의 명령어로 구분하였다.

표 4. 통신 프로토콜
Table 4. Communication Protocol

동작	Type	Length	Data(S/W, Key)
등록	0x30	0x02	0x05 0x11
인증	0x40	0x02	0x05 0x15
삭제	0x50	0x01	0x05

통신 프로토콜은 TLV(Type, Length, Value)방식을 따랐다[5]. 따라서 첫 번째 인자는 명령 구분을 위한 Type 데이터이며 두 번째 인자는 Value의 데이터 길이, 세 번째 인자는 명령에 따라 필요한 데이터 이다.

등록 동작에서는 시리얼 키를 등록하고자 하는 소프트웨어 또는 디지털콘텐츠의 고유번호와 시리얼 키 이며, 인증 동작에서는 인증하여야 할 소프트웨어 또는 디지털콘텐츠의 번호와 사용자의 PC에 설치된 시리얼 키 이다. 마지막으로 시리얼 키 삭제를 위한 명령은 삭제할 소프트웨어 또는 디지털콘텐츠의 번호만을 입력받아 시리얼 키를 스마트카드에서 삭제한다.

Ⅳ. 시스템 구현

본 논문은 자바카드 구현부와 MFC를 이용한 스마트카드 제어 프로그램 구현부로 구분하여 구현하였다.

4-1 구현 환경

아래의 표 5는 스마트카드를 이용한 시리얼 키 인증 시스템의 구현 환경을 보여주고 있다.

표 5. 구현 환경
Table 5. Implementation Environment

OS	Windows XP
Editor	Eclipse, Visual C++ 6.0
Library	JCOP, PC/SC

시스템의 구현을 위한 OS는 Windows XP를 사용하였으며 Editor는 자바카드 구현을 위한 Eclipse와 PC애플리케이션 구현을 위한 Visual C++ 6.0으로 구분하여 구현하였다. 또한 Eclipse에서는 자바카드 구현을 위하여 IBM에서 제공하는 JCOP(Java Card Open Platform)을 이용하였고, PC 애플리케이션에서 스마트카드 리더와의 통신을 위해서 PC/SC Library를 이용 하였다[6].

4-2 스마트카드 애플릿

스마트카드 애플릿은 PC에서 요청한 등록, 인증, 삭제 명령에 따른 스마트카드의 저장소에 데이터를 저장, 조회, 삭제를 하는 기능을 수행한다. 구현은 자바 언어 기반의 자바카드로 구현되며 기본 소스 코드는 아래와 같다.

표 6. 자바카드 소스 형태
Table 6. Form of JavaCard Source

```
import javacard.framework.APDU;
import javacard.framework.Applet;
import javacard.framework.ISO7816;
import javacard.framework.ISOException;

public class ngn2 extends Applet {

ngn2()
{
// 생성자
}

public static void install(byte[] bArray, short bOffset, byte bLength)
{
// 자바카드 인스톨
}
// 모든 APDU를 전달 받는다.
public void process(APDU apdu)
{
}
}
```

표 7. APDU 명령어 분기
Table 7. Division of Command APDU

```
// APDU 버퍼를 JCRE에서 전달받은 APDU 객체에서 가져온다.
byte[] buffer = apdu.getBuffer();

// return if the APDU is the applet SELECT command
if(selectingApplet())
return ;

// SELECT APDU 명령을 확인한다.
buffer[ISO7816.OFFSET_CLA] = (byte)(buffer[ISO7816.OFFSET_CLA] &(byte)0xFC);

if((buffer[ISO7816.OFFSET_CLA] == 0) && (buffer[ISO7816.OFFSET_INS] == (byte)(0xA4)))
return ;

// test1 애플릿에서 사용하는 CLA 명령을 검증한다.
if(buffer[ISO7816.OFFSET_CLA] != test1_CLA)
ISOException.throwIt(ISO7816.SW_CLA_NOT_SUPPORTED);

// 각각 정의된 APDU 명령어에 따라 private 함수 호출하도록 분기한다.
switch (buffer[ISO7816.OFFSET_INS])
{
case 0x30:
reg(apdu);
return ;
case 0x40:
certification(apdu);
return ;
case 0x50:
drop(apdu);
return ;
}
```

표 7 은 입력받은 APDU명령을 분석하여 등록, 인증, 삭제의 각 메서드로 분기하는 부분의 소스이다. 여기에서 분기된 결과에 따라 데이터를 분석하여 해당 시리얼 키를 등록, 삭제 또는 인증을 통한 인증성공 및 실패 메시지를 요청한 PC에 다시 전달하여 주는 역할을 한다.

4-3 클라이언트

클라이언트 애플리케이션에서는 소프트웨어가 실행될 때 스마트카드에 시리얼 키 인증을 요청하여 승인이 되면 실행하고, 인증에 실패하면 소프트웨어 실행을 중지하는 역할을 한다.

표 8 에서는 스마트카드의 연결을 위한 기본 동작의 소스를 보여주고 있다.

표 8. 스마트카드 연결 소스
Table 8. Connection Source of Smartcard

```
byte status[2];
DWORD dwStatusLength = sizeof(status);

// 자바카드에서 어플리케이션 선택을 위한 apdu

// 00 a4 04 00 byte수 aid 로 해야 선택 apdu

BYTE apdu[12] = {0x00, 0xa4, 0x04, 0x00, 0x07, 0xa0, 0x00, 0x00, 0x00, 0x00, 0x11, 0x10};

// 스마트카드 연결
SCardConnect(ContextHandle,pmszReaders,SCARD_SHARE_SHARED,SCARD_PROTOCOL_T0 | SCARD_PROTOCOL_T1, &hCardHandle, &dwAP);

// 전송 시작 준비
SCardBeginTransaction(hCardHandle);

// 스마트카드의 어플리케이션 선택
SCardTransmit(hCardHandle, SCARD_PCI_T1, apdu, sizeof(apdu), NULL, status, &dwStatusLength);
```

스마트카드와의 연결에 성공하면 클라이언트에서 구동 시 인증 명령을 스마트카드에 전송하고 시리얼 키의 등록은 설치 시, 그리고 삭제는 소프트웨어 삭제를 할 경우에 전송한다.

그림 7 은 클라이언트 시스템의 최종 구현 결과를 보여주는 것으로 스마트카드에 의하여 인증되어 소프트웨어가 실행되는 결과이다.

그림 7. 시스템 실행 화면
Fig. 7. System execution

V. 결 론

인터넷이 급격히 발전하고 인터넷 사용자의 수가 급격히 증가하면서 불법소프트웨어의 사용이 빈번히 일어나고 있다. 이에 소프트웨어와 디지털콘텐츠의 저작권을 보호하기 위해 스마트카드라는 보안성 높은 장치를 이용하였다.

스마트카드의 보안성은 이미 입증이 되어 있기에 프로토콜의 통신에 있어서 보안성에는 문제가 되지 않는다. 또한 스마트카드의 사용이 급격히 증가하고 있어 스마트카드 사용을 위한 리더기 보급은 전 국민적으로 일반화 될 것이다. 이에 스마트 카드를 이용한 시리얼 키 인증 시스템을 도입하게 되면 소프트웨어와 디지털콘텐츠의 저작권을 보호 할 수 있는 길이 열릴 것이다. 더불어 사용자의 측면에서도 스마트카드 한 장의 소지로 정당한 소프트웨어와 디지털콘텐츠의 이용을 할 수 있게 된다.

참 고 문 헌

[1] Michael R. Carr, "Smart Card Technology With Case Studies", *36th Annual 2002 International Carnahan Conference,* 20-24 Oct. 2002.

[2] T. Schaffer, A. Glaser, S. Rao, P. Franzon. “A ilp-chip implementation of the Data Encryption Standard(DES)”, *MCMC* ‘97 IEEE, 1997.

[3] "Java card technology at-aglance: The foundation for secure digital identity solutions", *SUN MICROSYSTEMS* INC, 2005.

[4] IBM, "OpenCard Framework 1.2 Programmer’s Guide ", *OpenCard Framework* 1.2, 1999.

[5] "Reserved Type, Length and Value (TLV) Codepoints in Intermediate System to Intermediate System", *RFC 3359*.

[6] Frank Seliger, "OpenCard and PC/SC - Two New Industry Initiatives for Smart Cards", http://opencard.org, 2002.

김 유 두 (金裕斗)

2007년 2월 : 한국기술교육대학교 인터넷공학(공학사)
2007년 3월~현재 : 한국기술교육대학교 대학원 정보미디어공학과 (석사과정)
관심분야 : 무선 네트워크, 모바일 콘텐츠, 스마트카드

문 일 영 (文日永)

2000년 2월 : 한국항공대학교 항공통신정보공학과 (공학사)
2002년 2월 : 한국항공대학교 대학원 항공통신정보공학과 (공학석사)
2005년 2월 : 한국항공대학교 대학원 정보통신공학과 졸업(공학박사)
2004년 ~2005년 : 한국정보문화진흥원 선임연구원
2005년 3월~현재 : 한국기술교육대학교 인터넷미디어공학부 조교수
관심분야 : 무선 인터넷 응용, 무선 인터넷, 모바일 IP