논문 2004-41CI-4-12

서명을 이용한 스마트카드 사용자 인증을 위한 COS 설계

(Design of COS for smart card user authentication using signature)

송 영 상*, 신 인 철**

(Young-Sang Song and In-Chul Shin)

요 약

본 논문에서는 스마트카드의 사용에서 보편적으로 사용되는 패스워드 대신 서명을 이용하여 사용자 인증 시스템을 구현하였다. 서명은 사용자에게 익숙하여 특별한 기억이 필요 없으며, 강제에 의한 유출로 야기되는 타인사용의 가능성이 어려워 패스워드보다 안전하다. 그러나 서명 데이터의 크기는 매우 커 스마트카드에서 처리하기 위해 특별한 명령어 필요하며, 이를 위해 ISO 7816-3, 4의 표준에 따른 기본 명령어와 사용자 인증과 서명 데이터를 처리하기 위한 명령어로 스마트카드의 운영체제인 COS를 설계 구성 하였다. 또한 사용자, 카드, 단말기 및 서명 DB서버 사이의 프로토콜을 설계하였다. 서명 데이터는 등록 과정에서 서명 DB서버와 스마트카드에 저장되며, 사용자 인증은 사용자가 입력한 서명 데이터와 서명 DB에 저장된 데이터로 비교 알고리즘을 수행하여 인증 결과와 서명 데이터의 해쉬 값을 카드 쪽에 전송하여 스마트카드 사용자 인증이 이루어진다. 이 과정에서 서명 DB서버와 사용자 간의 상호 인증도 이룰 수 있다. 본 논문에서 제시한 시스템은 사용자 서명과 스마트카드 내의 서명 데이터를 비교하여 상호간의 신뢰성을 보장 받을 수 있으며, 사용자에게 좀 더 안전하고 간편한 서비스를 제공할 수 있을 것으로 기대된다.

Abstract

This paper suggests the way to realize smart card security system by using handwritten signature instead of a password which is traditionally used for user authentication. Because of the familiarity of signature we don't need to try to remember the password and signature is difficult to be used by guess or illegal forced situation. The feature data of handwritten signature is large, so we designed COS which is consist of special commands for processing user's handwritten signature data, user authentication, and basic commands based on ISO 7816-3. Also protocol among user, smart card, terminal and DB server is designed. In registration process, the feature data of user signature is saved in both a DB server and a smart card. User authentication is processed by comparing the user signature and the saved feature data in a smart card and in a DB server. And the authentication result and hash value of signature data in DB server are transferred to smart card. During this process the authentication between DB server and user is finished. The proposed security system has more higher level of security in user authentication of smart card and it will provide safer and more convenient security services.

Keywords: Smart card, Signature, User Authentication, COS(card operating system)

I. 서 론

인터넷의 폭발적인 확산과 함께 정보처리 기술의 빠른 발전으로 언제, 어디서나 누구와도 통신이 가능할 뿐만 아니라 네트워크를 통해 가정이나 사무실에서의 업무처리 및 전자 상거래가 보편화되어 가고 있다. 이로써 온라인 사업이 증가하고, 고객에게는 다양한 서비스가 제공되고 있지만 보다 안전하고 신뢰성 있는 서비스의 제공 및 통신을 위해서 정보보호는 필수적이며 또한 개인의 정보보호 및 사용자 인증에 대한 관심이 높아져 갈 뿐만 아니라 많은 연구가 이루어지고 있다.[1-4]

최근에는 스마트카드가 개인의 정보보호 및 사용자 인증 시스템에 널리 적용되고 있다. 스마트카드는 일반

* 종신회원, 단국대학교 대학원 전자컴퓨터공학과
(Department of Electronic and Computer Engineering, Graduate School, Dankook University)
** 종신회원, 단국대학교 전기전자컴퓨터공학부
(Department of Electrical, Electronic and Computer Engineering Dankook University)
※ 이 연구는 2000년도 단국대학교 대학연구비의 지원으로 연구되었음.
접수일자: 2003년12월24일, 수정완료일:.2004년7월7일

플라스틱 카드에 마이크로프로세서와 메모리 시스템을 내장하고 있는 칩이 있으며 물리적인 조작에 의해서는 외부에서 데이터를 획득할 수 없도록 제작되어 있어 자체적인 계산 능력 및 데이터 저장 능력과 함께 뛰어난 보안성을 갖고 있으며, 카드의 메모리에는 운영체제가 내장되어 있다. COS(card operating system)는 파일 관리, 데이터 통신, 보안 시스템을 내장하여 암호 알고리즘 수행, 데이터 무결성, 사용자 인증, 단말기와 카드간의 상호 인증을 위한 보안 모듈을 가지고 있는 펌웨어 프로그램이라 할 수 있다. COS는 각각의 활용분야에 따라 응용 프로그램을 작성할 수 있으며 개인 신분 증명, 접근제어, 금융 분야 및 여러 응용 분야에서 활발히 사용되고 있다.[4,5,8,9,10] 기존 스마트카드는 카드의 사용자 확인을 위해 보편적으로 8Byte의 크기의 패스워드, 즉 PIN(personal identification number)을 사용한다. 그러나 패스워드의 기억에 따른 불편, 강제에 의한 패스워드의 유출 및 추측에 의한 타인사용 가능성 등에 따른 위험이 있다. 이러한 사용상의 불편과 위험을 감소시키고 좀 더 안전성 있는 사용자 인증을 위해 생체 정보를 이용한 사용자 인증 시스템이 적용되고 있는 추세이며, 생체 인식은 홍채, 얼굴, 지문 등 신체적 특징을 이용한 방법과 서명, 발걸음, 타이핑 습관 등 습관적 특성을 이용한 사용자 인증에 대한 연구가 활발히 진행되고 있다.[1,2,8,19,20]

본 논문에서는 일반적으로 사용에 거부감이 없고 친숙하며 타인에 의한 도용이 어렵고 강제에 의한 유출과 타인 사용이 불가능한 서명을 이용하여 스마트카드 사용자 인증을 위한 COS를 설계하였다. 서명 데이터는 온라인 서명데이터를 사용하였으며 서명 데이터는 파라메터의 선정과 적용 알고리즘에 따라 달라지기는 하나 보통 수 KB정도이다. 서명 데이터는 온라인 서명데이터를 사용하였으며 서명 데이터는 파라메터의 선정과 적용 알고리즘에 따라 달라지기는 하나 보통 수 KB정도이다. 서명 데이터를 보관하고 인증시 입력 서명 데이터를 받아드려 파일 형태로 저장, 비교하여 처리하므로 파일 처리 및 비교를 위한 명령어가 설계되어야 한다. 또한 8byte의 PIN과는 달리 서명 데이터는 2-4Kbyte 정도로 크고 스마트카드의 최대 입출력 데이터 전송 단위가 80byte 크기로 처리된다. 이를 위해 명령어에 오프셋을 지정해 가면서 데이터를 전송하여 카드(EEPROM)에 저장하기 위한 명령어 설계가 필요하다.

설계된 시스템을 실제 운영하고 확인하기 위하여 단말기와 카드의 데이터 전송 프로토콜(T=0) 및 명령어(APDU)를 표준에 따라 스마트카드의 운영체제인 COS를 설계 및 구현하여 확인하였다. 사용자 등록은 서명 DB서버에 사용자가 자신의 서명을 제출하면, 서명으로부터 특징 파라미터를 추출하여 서명 DB를 구축함과 동시에 서명 처리 명령어를 이용하여 스마트카드에 서명 데이터를 저장하고 사용자에게 스마트카드를 발급하는 시나리오를 가정하였다. 스마트카드를 받은 사용자는 자신이 지니고 있는 카드와 사용 시 입력하는 서명으로 사용자 인증을 받고 그 과정에서 자신이 이용하는 서명 DB서버를 인증함으로써 안전하고 편리하게 서비스를 제공받을 수 있다.

II. 시스템 구성 및 스마트카드

1. 시스템 구성

서명 인증 시스템은 전자펜 또는 stylus펜(PDA용)을 이용하여 입력된 개인의 서명 특성을 검증하는 것으로써 서명의 특징 즉 모양, 속도, 필압, 획 순서 등의 정보를 통합하여 비교 분석하여 본인 여부를 확인할 수 있다. 이때 구성되는 서명데이터는 보통 수KB의 크기를 가지며 사용되는 파라메터의 종류와 양에 따라 비교 속도 및 결과에 영향을 미친다. 인증시스템은 크게 등록과정과 인증 과정으로 구성된다.[19]

등록과정은 사용자가 자신의 ID와 함께 서명을 수차례를 제출하며, 제출된 서명에서 처리 과정을 통해 특징을 추출(feature extraction)하여 개인의 고유한 특징 파라미터를 통합한 데이터를 등록한다. 이를 동시에 서명 DB서버와 스마트카드에 저장하여 등록과정을 끝마치게 된다.

사용자 인증 과정에서는 사용자가 제출하는 서명을 받아 특징을 추출한 후 서명 DB서버에 등록된 서명데이터와 비교가 이루어진다. 이때 두 개의 데이터의 유사도 (similarity)를 비교하기 위해 거리측정법(distance measure), 뉴럴 네트워크 (neural network), DP 매칭 등의 방법을 사용하여 정해진 임계 값보다 작으면 사용자를 인증이 이루어지고, 그렇지 않으면 사용자 인증이 거부된다. 그림 1.에 서명등록 및 인증 과정을 보였다. 점선은 사용자 등록과정을 나타내고 실선이 사용자 인증 부분을 나타내고 있다. 사용자의 서명 데이터 추출과 비교 알고리즘은 상용제품의 프로그램을 이용하여

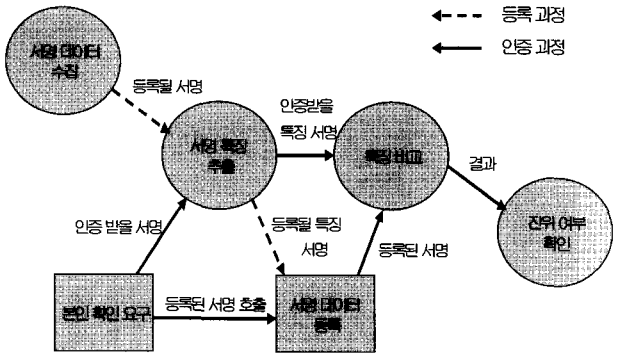


그림 1. 서명 인증 시스템
Fig. 1. Signature authentication system.

본 연구에 적용하였다.

2. 스마트카드

(가) 스마트카드

스마트카드는 신용카드 크기의 플라스틱 판 위에 IC 칩을 장착한 카드이다. IC칩은 CPU와 카드를 운영할 수 있는 운영체제인 COS가 내장된 ROM, 특정한 데이터 및 키를 보관하는 EEPROM, 데이터 연산을 위해 필요한 RAM, 보안 시스템을 위해 특정한 암호 알고리즘을 계산하는 코프로세서(coprocessor)가 내장되어 있으며 외부 인터페이스를 위한 IC칩의 접촉판으로 구성되어 있다.[4-10]

COS는 기본적으로 파일을 선택하고 생성하는 파일 관리 기능과 인증, 암/복호화 등의 보안 기능, 외부 단말기와의 통신 기능 등을 가지고 있다. COS는 내장된 CPU의 명령어로 프로그래밍 된 펌웨어(firmware)의 일종이다. COS에는 보안성을 유지하기 위한 여러 가지의 보안 시스템들이 내장되어 있다. 스마트카드의 보안 메커니즘은 다음과 같다.[5,6,8]

- 사용자 인증(user authentication) : PIN입력
- 파일 접근제어(access control)
- 세션키(session key) 생성 : 카드 내의 키(key) 노출을 피하기 위함
- 내부 인증(internal authentication) : 터미널이 카드를 인증
- 외부 인증(external authentication) : 카드가 터미널을 인증
- 데이터 암호화 및 복호화 (encryption/decryption)
- MAC(message authentication code)생성 : 데이터의 무결성 보장을 위한 데이터 인증 코드

본 논문에서 사용한 삼성 S3C89K8칩으로 블록도를

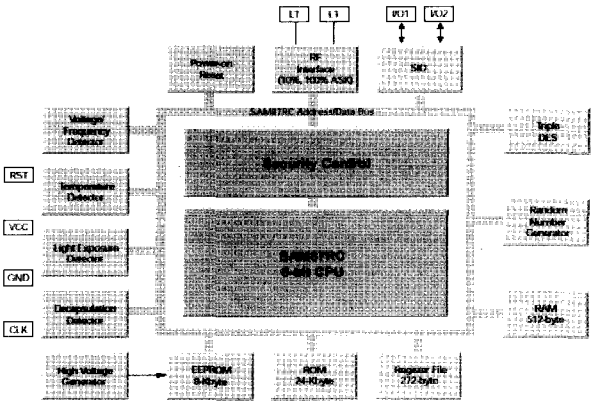


그림 2. S3C89K8 블록도
Fig. 2. S3C89K8 block diagram.

표 1. 명령어 APDU의 내용
Table 1. Command APDU structure.

필수 헤더				가변길이		
CLA	INS	P1	P2	Lc	Data	Le

코드	내용	길이 (byte)
CLA	Class of instruction	1
INS	Instruction Code	1
P1	Instruction Parameter 1	1
P2	Instruction Parameter 2	1
Lc	명령어 데이터 필드 내에 존재하는 바이트 수	0또는1
Data	명령어로 보내는 데이터 바이트	가변
Le	응답 데이터 필드에서 예상되는 최대 바이트 수	0또는1

그림 2.에 보였다. 스마트카드의 접점은 총 8개로 구성되며, 8bit의 CPU와 Triple-DES를 수행하기 위한 coprocessor 및 난수 발생기 등을 가지고 있으며, I/O와 RF 통신이 가능한 마이크로 컨트롤러로 구성된다. 또한 내부에 COS가 들어가는 24Kbyte의 ROM과 내부 데이터 연산을 위해 사용되는 512byte의 RAM과 272byte의 Register, 파일에 관련된 데이터를 저장하는 8Kbyte의 EEPROM으로 구성되어 있다.

(나) 명령어

명령어는 COS의 기능과 밀접한 관련이 있다. 카드와 단말기간의 호환성 확보 등을 위하여 ISO 7816를 참조하여 명령어가 구현된다.

메시지 전송 프로토콜(T=0)에 따라 카드와 단말기간의 전송이 이루어진다. T=0 프로토콜은 바이트 단위로 데이터를 단말기와 응용 프로그램간의 데이터를 주고받기 위한 전송 프로토콜로 현재 가장 많이 사용된다. 단말기 및 카드는 물리적 계층, 데이터링크 계층, 전송 계

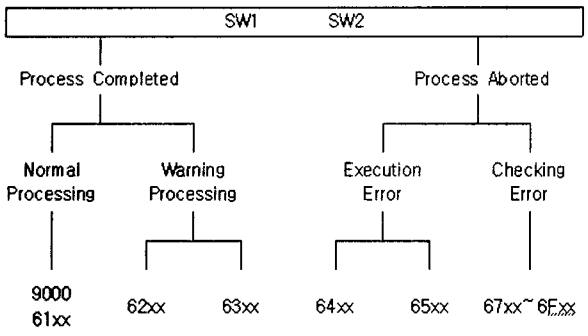


그림 3. 상태 바이트의 구조 설계
Fig. 3. Structure scheme of status bytes.

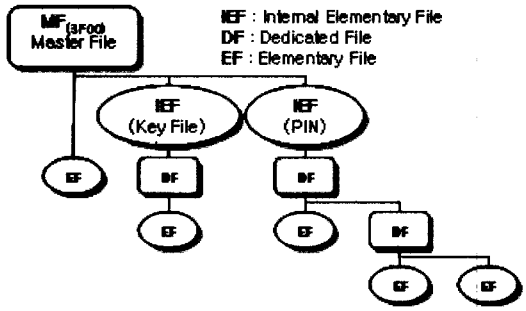


그림 4. 파일 시스템
Fig. 4. File System.

층을 수행한다. 어플리케이션의 실행을 위해서는 추가 계층인 어플리케이션 프로토콜이 단말기에서 수행되어야 하는데 어플리케이션 프로토콜의 수행 단계는 카드에 명령어 전송, 카드에서의 명령어 처리 및 명령어에 대한 카드 응답으로 구성된다. 따라서 특정 응답은 특정 명령어와 일치하여야 하며, 명령어-응답 쌍으로 표시 되어진다. 이를 위해 APDU(application protocol data unit)는 카드에서 터미널 또는 터미널에서 카드로 전송되는 명령어 메시지나 응답 메시지를 포함하며, 명령어-응답 쌍내에 존재하는 명령어 메시지 및 응답메시지는 데이터를 포함할 수 있다. 명령어 APDU는 표 1.에서 보는 것처럼 4바이트의 필수 헤더 및 가변길이의 Conditional Body로 구성된다.

(다) 상태 응답(SW : status word)

명령어 처리의 상태를 표시하는 바이트 SW1, SW2는 응답메시지로 전송 계층에 의하여 응용계층으로 전송되고, 상태 바이트의 코딩 및 코딩구조는 그림 3.과 같다.[5,6]

(라) 스마트카드 파일 시스템

스마트카드의 파일 시스템을 그림 4.에 나타내었고

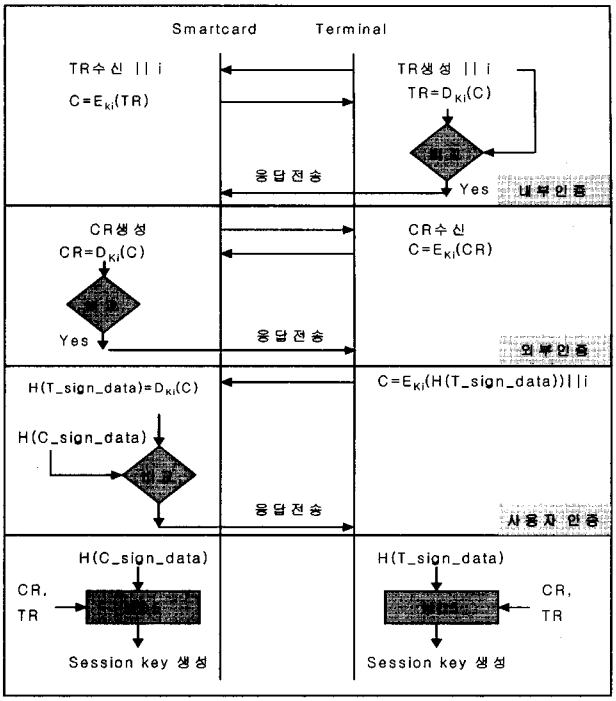


그림 5. 스마트카드의 보안 메커니즘
Fig. 5. Smart card security mechanism.

이것은 DOS의 계층적인 파일구조와 유사하다.

MF(master file : 3F00)은 가장 상위 계층으로 루트 디렉토리에 해당되며, DF (dedicated file)는 MF의 하위에 존재한다. EF(elementary file)은 디렉토리 내의 파일에 해당되며 EF는 Internal EF, working EF로 구분된다. 이곳은 카드에 사용되는 데이터 중 보안성을 유지해야 하는 데이터 및 어플리케이션에 사용되는 데이터를 저장하는 장소로 사용된다.[5,6]

Ⅲ. 서명을 이용한 스마트카드 사용자 인증 설계

1. 스마트카드 인증 프로토콜

일반적으로 스마트카드의 인증은 카드와 단말기 간의 상호 인증인 외부인증(external authentication)과 내부인증(internal authentication)이 이루어 진 후 사용자 인증이 이루진다. 인증과정이 정상적으로 이루어지면 카드와 단말기간의 통신에 사용될 세션키를 생성한다. 세션키는 상호간의 데이터의 기밀성을 유지하기 위해 사용되며, 세션키 생성은 CR, TR, 공유된 key에 의해서 생성된다. 단말기에 카드를 삽입하면 카드로부터 ATR(answer to reset)을 받고 카드는 외부로부터 명령을 받기 위한 상태가 된다. 위의 과정은 그림 5.와 같이 총 4단계로 구성되어 있으며 카드와 단말기간의 통신상

의 보안은 상호 공유하고 있는 키 테이블의 키들 중 하나인 Ki를 이용한다. 구체적 사항은 다음과 같다.

(가) 내부인증(Internal Authentication)

내부인증은 단말기가 사용되고 있는 카드를 인증하기 위한 절차이며 절차는 다음과 같다.

① 단말기는 자신의 TR을 생성하여 사용할 암호화 키 번호(i)와 함께 카드로 전송한다.

② 카드는 TR을 수신하여 사전 공유된 키(Ki)로 암호화하여 단말기에 재전송 한다.

$C = E_{Ki}(TR) \| i$ (E : encryption)

③ 단말기는 C를 수신하여 C를 복호화한 후 자신이 보낸 TR과 복호화 된 TR이 같은지 비교하여 카드를 인증한다.

$TR = D_{Ki}(C)$ (D : decryption)

(i = key number)

(나) 외부 인증(External Authentication)

외부인증은 카드가 단말기를 인증하기 위한 절차이며 절차는 다음과 같다.

① 카드는 자신의 CR을 생성하여 단말기에 전송한다.

② 단말기는 CR을 수신하여 Ki로 암호화한 후 카드에재 전송한다.

$C = E_{Ki}(CR)$

③ 카드는 C를 수신하여 C를 복호화한 후 자신이 보낸 CR과 복호화 된 CR이 같은지 비교하여 카드를 인증한다.

$CR = D_{Ki}(C)$

2. 서명을 이용한 사용자 인증 설계 및 세션키 생성

사용자 인증은 사용자가 정상적으로 등록이 되어진 후 발급된 카드의 사용자 인증을 확인하는 절차이며 내부 및 외부 인증이 완료 된 후 이루어진다. 인증 절차는 다음과 같이 설계 하였다.

사용자가 제출한 서명을 DB서버에 저장되어 있는 데이터와 비교한 다음 카드로 사용자 인증 결과 와 함께 서명 데이터의 해쉬 값(H(T_Sign_data)을 전송한다. 카드는 이를 받아 카드에 저장되어 있는 서명 데이터를 해쉬를 수행한 값(H(S_Sign_data)과 수신된 해쉬 값을 비교하여 사용자 인증이 이루어진다.

① 사용자가 입력한 ID와 사용자의 서명을 서명 DB 서버로 전송한다.

② 서명 DB서버에서 사용자의 서명으로부터 특징 추출하여 비교 알고리즘을 통해 사용자 인증이 이루어진다.

③ 그에 따른 인증 결과와 서명 해쉬 값을 카드로 송신하며, 스마트카드는 그 데이터를 수신한다. 이때 데이터는 공유된 키(Ki)로 암호화 하여 전송한다.

$C = E_{ki}(H(T_Sign_data)) \| i \| $사용자 인증 결과

④ 수신된 데이터 중 사용자 서명에 대한 응답이 OK 이면, 다음으로 스마트카드의 EEPROM에 파일 3F05을 생성하여 서버로부터 수신된 해쉬 데이터를 저장한다. 카드에서는 3F06의 파일을 생성하여 카드에 저장되어 있는 서명 데이터를 해쉬 함수를 통한 결과 값을 저장한다. 그리고 Sign_Compare 명령어를 이용하여 두 개의 데이터가 동일할 때 서버를 인증 확인한다. Sign_Compare 명령어는 COS의 Command이다.

Verify = 사용자 인증 결과 검색

$H(T\_Sign\_data) = D_{ki}(C)$

Sign_Compare = H(C_Sign_data), H(T_Sign_data)

⑤ Session Key 생성 : 이과정은 단말기와 스마트카드간의 데이터 통신을 위한 세션키 생성을 하는 과정으로 단말기로부터 수신한 TR과 카드에서 생성된 CR, 카드에 저장된 서명 데이터를 통해 세션키를 생성한다. 본 논문에서 사용한 스마트카드 타겟 보드는 S3C89K8로 Triple DES를 지원하며, 통신 암호화를 위해서 사용되는 키이다. 서버 쪽도 마찬가지로 스마트카드의 CR을 수신하여 동일한 방법으로 세션키를 생성한다. Ks는 16byte의 생성된 세션키 이다.

카드 쪽 : Ks = Hash(H(T_Sign_data), CR, SR)

단말기 쪽: Ks = Hash(H(C_Sign_data), CR, SR)

3. COS 명령어 설계

본 연구에서 이용한 스마트카드의 COS에 설계된 명령어는 ISO 7816의 표준에 따른 기본 명령어와 큰 서명 데이터를 처리하기 위해 설계한 명령어로 구분 된다. 설계된 명령어 중 Sing_Compare, MD5는 추가 설계된 명령어며, 나머지 명령어는 ISO 7816 표준에서 제공되는 명령어를 본 연구에 적절한 형태로 수정 설계 하였다.

표 2. 명령어 코드 설계
Table 2. Commands code design.

구분	Command name	Value	설 명
기본 명령어	Read Binary	B0	EF 파일의 내용을 읽기
	Update Binary	D6	APDU에 주어진 데이터를 EF 파일에 업데이트하는 명령어
	Get Data	CA	카드로부터 필요한 정보를 읽어 오는 명령어
	Put Data	DA	카드관련 정보를 카드의 특정 영역에 기록하는데 사용
	Select File	A4	파일을 선택하는 명령어
	Internal Auth.	88	단말기로부터 전송된 challenge 데이터와 카드에 저장된 정보를 사용하여 인증 절차를 수행하는 명령어
	External Auth.	82	카드에 의해 사전에 발생된 Challenge와 카드에 저장된 key 및 단말기에 의하여 전송된 인증데이터를 근거로 하여 검증한 결과에 따라 보안 상태를 업데이트하는 명령
	Get Response	C0	전송하지 않은 APDU를 카드에서 단말기까지 전송하기 위해 사용되는 명령어 즉 카드의 응답을 강제로 읽어 오는 명령어
	Get Challenge	84	Challenge의 발생을 요구 하는 명령어
설계된 명령어	Create File	E0	APDU에 주어진 파일 이름으로 EF 파일 생성
	Create Session	8A	카드와 단말기간의 데이터 통신에 사용할 key 생성 명령어
	Verify	20	사용자 인증 결과를 받는 명령어
	Sign_Compare	CE	단말기로부터 받은 서명데이터와 카드내의 서명데이터를 비교하는 명령어
	MD5	22	해쉬 함수를 수행하기 위한 명령어

표 2.는 스마트카드의 동작을 위해 사용되어 지는 명령어로 표준에 따른 기본 명령어와 서명데이터를 처리하기 위한 설계된 명령어를 나타내고 있다.

4. 서명 데이터 등록

사용자의 서명에서 특징 추출된 데이터를 데이터베이스와 스마트카드에 저장 한다.

서명 데이터는 2KByte의 크기를 갖는다. 스마트카드에서는 MF(3F00) 밑에 3F03의 EF을 만들어 저장한다. 단말기에서 카드 쪽으로 서명 데이터를 한 번에 최대로 보낼 수 있는 크기는 80byte이다. 최대 파일 크기를 2KByte를 가질 수 있게 스마트카드의 메모리 맵을 설정하여 16번을 연달아 데이터를 전송하여 사용자의 서명데이터를 스마트카드에 저장하여 사용자 등록이 이루어진다. 저장되는 단계는 다음과 같다.

· Create File : EEPROM에 서명 특징 데이터를 저장을 위해 3F03의 EF파일을 생성하고, 파일의 크기를 2K로 지정해 주었다. 데이터의 총 길이는 06byte이고, 데이터의 왼쪽에서부터 보면 3F03이 생성될 EF이고 0000는 파일 접근제어에 대한 권한을 줄 수 있다. 마지막 뒤의 2byte는 파일크기를 표시하고 있다. 본 논문에서는 최대 파일 사이즈를 0800까지 즉 2Kbyte까지 기록할 수 있도록 COS를 설계하였다.

CLA	INS	P1	P2	Lc	Data
90	E0	00	00	06	3F0300000800

· Select File : 스마트카드 파일 시스템에서 EF파일을 선택하는 명령어이다.

CLA	INS	P1	P2	Lc	Data
00	A4	00	00	02	3F03

· Update File : Update File은 한번에 전송할 수 있는 데이터 크기가 80Byte로 2Kbyte를 16번에 걸쳐 전송한다. 전송되는 데이터를 메모리에 쓰기위해 offset의 위치결정은 P1, P2에 지정하여 사용하였다.

CLA	INS	P1	P2	Lc	Data
00	D6	00	00	80	저장될 데이터

5. 사용자 인증

(가) 내부 인증

단말기가 생성한 TR과 1바이트 i를 카드에 전송하고 카드는 i로 선택된 키로 TR을 암호화 하여 단말기에 전송한다. 다음은 내부인증을 수행하기 위한 명령어를 보여 주고 있으며, 데이터의 마지막 1byte(Le)는 키의 선택 값을 표시하고 있다.

CLA	INS	P1	P2	Lc	Data	Le
00	88	00	00	TR 길이+1	TR ‖ i	1

(나) 외부 인증

카드가 생성한 CR을 단말기로 전송하여 외부인증이 이루어진다.

CLA	INS	P1	P2	Lc	Data
00	82	00	00	CR 길이	CR

(다) 사용자 인증

기존의 스마트카드는 응용프로그램과 카드와의 인증인 내부/외부 인증만 이루어지나 본 논문의 시스템은 생체 정보를 이용하여 DB서버와 카드와의 상호 인증이 이루어짐으로써 좀 더 보안성이 높은 시스템이라 할 수 있다. 그림 6.에서 보듯이 서버 단에서 사용자 인증이 이루어지면 사용자 인증 확인 데이터를 받아 처리 한

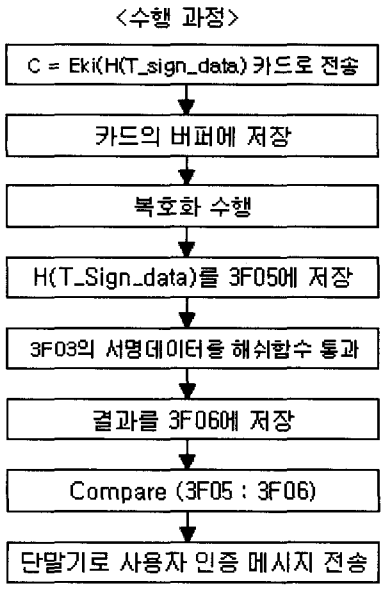


그림 6. 서명 데이터를 이용한 사용자 인증 처리 과정
Fig. 6. Using signature for processing user authenti -cation in smart card

다. 다음 명령어가 정상적으로 수행되어 지면 스마트카드의 사용자 인증을 위한 단계를 수행한다. 만일 수행되지 않으면 사용자 인증 확인 데이터를 재 요구 하고 3번을 넘기면 카드는 잠기게(Unblock) 된다.

· Verify : 사용자 인증을 수행하기 위한 Command 이다. 서버부터 받는 데이터가 카드에 저장된 값과 비교하여 같으면 파일 생성 명령어를 실행한다.

CLA	INS	P1	P2	Lc	Data
90	8A	00	00	08	사용자 인증 확인 데이터

· Create File : 서명 DB서버로부터 가져올 해쉬 값을 저장할 파일을 생성

CLA	INS	P1	P2	Lc	Data
90	E0	00	00	06	3F0500000016

· Update Binary : DB에 저장되어 있는 서명 데이터를 해쉬 함수인 MD5를 통해 서명 해쉬 함수(H(T_Sign_data)를 만들고 이 데이터를 Ki로 암호화 하여 카드에 전송한다. 다음의 데이터가 전송되면 복호화를 하여 위에 생성한 EF(3F05)에 저장한다.

CLA	INS	P1	P2	Lc	Data
00	B0	00	00	16	E_{Ki}(H(T_Sign_data))

· MD5 : 카드에 저장되어 있는 서명데이터를 해쉬 함수를 수행된 결과를 EF(3F06)에 저장한다.

CLA	INS	P1	P2	Lc	Data
00	22	00	00	02	3F03

· Sign_Compare : 서명 DB서버로부터 가져온 해쉬 값(3F05)과 카드에서 수행한 해쉬 값(3F06)을 비교하여 사용자 인증 및 서버 인증이 이루어진다.

CLA	INS	P1	P2	Lc	Data
00	B4	00	00	04	3F053F06

(라) 세션키 생성

· Create Session : 카드의 메모리에 저장되어 있는 서명 해쉬 데이터(3F06)와 카드의 난수(CR)와 터미널의 난수(TR)을 이용하여 Hash함수를 통해 최종적으로 128bit의 키를 생성한다. 세션키 생성 APDU는 90 8A 00 00 이다. 생성된 키는 RAM에 저장되어 통신이 끊어질 때 까지 사용된다.

Ⅳ. 구현 및 실험

1. 개발 환경

본 논문에서 개발을 위한 환경으로 크게 하드웨어와 소프트웨어로 나눌 수 있다. 하드웨어로는 스마트카드 Emulator, Target Board, Reader, Tablet이 있으며, 소프트웨어로는 COS 설계를 위한 프로그램, 서명을 처리하기 위한 프로그램, 어플리케이션을 프로그래밍하기 위한 프로그램으로 나눌 수 있다. 각 부분은 다음과 같다.

(1) 스마트카드 : 삼성 S3C89K8 Device
(2) 개발 환경 : PentiumIII Processor, Windows 2000 Server
(3) 개발 언어 : Assembly, Borland C++ Builder 5.0

스마트카드의 COS 구현을 위해서 OPENice i500에서 제공되는 프로그램을 이용하여 어셈블리어로 프로그래밍 하고, 어플리케이션 프로그램을 위해 Borland C++ Builder 5.0을 이용하여 프로그래밍 하였다.

(4) 개발 도구 : OPENice i500 Emulator, COS Tester, Target Board(S3C89K8), Reader(PC/SC 지원)

그림 7.은 프로그래밍 된 COS의 동작을 테스트 하는 프로그램으로 명령어 및 데이터를 카드에 전송하여 올바르게 작동하는지 검증 할 수 있는 프로그램이다. 마찬가지로 Serial port로 연결되어 있으며, 명령어 하나

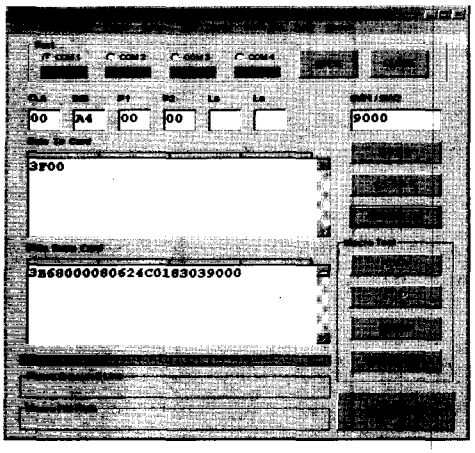


그림 7. 명령어 테스트 프로그램
Fig. 7. APDU Tester Program.

표 3. COS 설계을 위한 파일
Table 3. File for COS design.

파일이름 (.src)	파 일 설 명
ConstEqu	Constants를 정의
E2promEq	EEPROM의 Map을 정의
RAMequ	Internal/External 메모리의 Map을 정의
Main	COS 동작의 Main Routine
RcvByte	COS의 I/O와 Interrupt를 정의, 수신되는 데이터의 처리 루틴
SendByte	COS의 I/O를 통해 데이터를 송신하는 루틴 정의
Write	Smart card의 메모리에 데이터를 쓰기 위해 정의
CrytpUt	암호화에 사용되어지는 Utility 정의
DES	Triple DES를 사용하기 위해 DES 프로그램 소스
MD5	Hash인 MD5를 사용하기 위한 프로그램 소스
Command	COS에 사용되어지는 명령어들의 모음 (ISO 7816-3, 4 및 응용 Command 등)
Error	모든 명령어 및 데이터의 Error를 정의한 파일

씩 전송할 수 있을 뿐만 아니라 파일 형식으로도 전송할 수 있다. 스마트카드의 가상의 동작 시나리오를 확장자 .tst 의 파일로 저장하여 테스트 하면 그에 따른 결과를 동일한 이름으로 .rst 확장자 파일을 생성해주고, 가상의 시나리오에 따른 올바른 동작이 되었는지 살펴 볼 수 있다. 표 3.은 OPENice i500을 이용하여 COS 구현을 위해 설계된 소스 파일로 데이터 입출력, 카드의 메모리 할당, 보안관련 및 명령어에 관련된 소스들로 구성된다.

(마) 동작 확인

어셈블리어로 구현한 COS를 테스트하기 위해 어플리케이션 프로그램을 프로그래밍 하여 확인하였다.

그림 8.은 COS의 코드 중 서명 데이터를 처리 하는 부분을 위주로 프로그래밍 하였다. 서명을 처리 하는

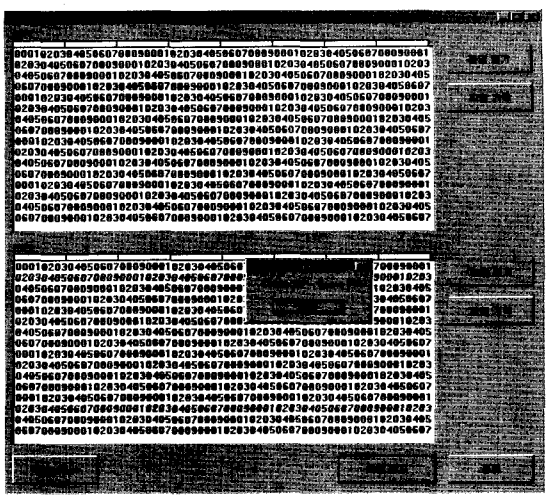


그림 8. 명령어 확인 응용 프로그램
Fig. 8. Application Program of Verify Command.

COS의 명령 루틴 코드는 1Kbyte의 코드 크기를 가지고 있으며 많은 데이터를 빠르게 처리하기 위해 내부의 8bit 레지스터를 이용하여 프로그래밍 하였다. 프로그램은 명령어를 카드에 전송하여 동작확인을 하기위한 응용 프로그램이다. 프로그램 동작은 Connect 버튼을 눌려 카드와 응용 프로그램과의 통신을 위해 단말기 정보 및 초기화작업을 수행한다. 이때 카드로부터 전송 받은 ATR을 표기 하고, 카드에 내부/외부 인증 서비스가 이루어진다. 사용자 인증을 위해 서명데이터를 카드에 update 한다. 파일 비교 버튼은 카드에 저장되어 있는 파일과 비교하여 파일이 일치 할 경우 카드로부터 받은 응답을 표시하고, 1bit라도 차이가 있으면 오류를 발생한다.

V. 결론 및 고찰

흔히 스마트카드의 사용자 확인을 위해 PIN 또는 패스워드가 사용된다. 그러나 이러한 시스템은 패스워드의 망각 또는 추측 및 강제에 의한 불법 사용의 가능성을 배제 할 수 없다. 이러한 불편과 위험성을 최소화시키고 일반적으로 사용이 친숙한 생체정보인 서명을 이용하여 스마트카드의 사용자 인증을 위한 COS를 설계 구현하였다.

서명데이터는 파라메터의 선정과 사용 알고리즘에 따라 데이터 량이 달라지나 흔히 사용되는 8바이트의 패스워드와는 스마트카드 상에서 구현방법이 달라진다. 서명을 이용한 사용자 인증 시스템을 스마트카드에서 구현하기 위해 서명 데이터를 받기 위한 Tablet과 그

과정은 상용 어플리케이션 프로그램을 사용하였고, COS 개발을 위해 OPENice i500 Emulator와 Target Board로는 삼성 S3C89K8을 이용하여 설계하고 구현하였다. Target Board의 EEPROM Size는 8Kbyte인 관계로 서명데이터가 2Kbyte를 넘어가면 안되는 조건으로 프로그래밍 하였다.

이를 처리하기 위하여 ISO 7816 표준에 따라 기본 명령어와 응용 명령어를 설계하였고, 등록, 인증, 상호 통신을 위한 루틴을 구현 하였다. 서명 데이터는 2KB의 크기를 가지며, 서명 DB서버 및 스마트카드의 EF에 저장 하였다. 사용자가 제출하는 서명은 서명 DB서버에서 특징 파라미터 추출 및 비교 알고리즘을 수행하여 인증되며, 인증이 이루어지면 서명 DB서버에 저장되어 있는 서명데이터의 해쉬 값을 카드에 보내고 카드는 이 값을 조사하여 서명 DB서버를 인증하며 사용자, 서명 DB서버, 스마트카드의 상호 인증이 동시에 이루어지게 된다. 모든 명령어는 프로토콜 수행 결과 표준에 따른 상태응답에 맞게 프로토콜이 이루어 져 있어 온라인과 오프라인에서도 사용가능 할 수 있을 것이다.

본 연구 결과는 스마트카드에서 서명데이터를 이용한 보안 시스템 구성을 위한 기본 사양 설계와 구현에 적용시켜 좀 더 다양한 응용분야의 개발에 응용될 수 있을 것이다. 또한 다양한 생체정보에 확대하여 적용할 수 있을 뿐 아니라 이를 이용하여 사용자는 좀더 안전하고 편한 환경에서 온라인 및 오프라인 서비스를 이용할 수 있을 것이다. 서명 비교 알고리즘의 수행을 스마트카드내에서 수행하기 위한 COS 개발은 추후 연구로 미루었다.

참 고 문 헌

[1] Luca Bechelli, Stefano Bistarelli, Anna Vaccarelli, "Biometrics authentication with smart card", http://www.iat.cnr.it/attivita/progetti/parametri_biomedici.html

[2] Gael Hachez, Francois Koeune, Jean-Jacques Quisquater "Biometrics, Access Control, smart card : A Not so simple combination", http://citeseer.nj.nec.com/cs

[3] Giampaolo Bella, "Modelling Security Protocols Based on Smart Cards", http://citeseer.nj.nec.com/cs

[4] C.P. Schnorr, "Efficient identification and signatures for smart card", Advances in Cryptology Crypto'89, Lecture Notes in Computer Science, G. Brassard(ed.), Berlin Springer-Verlag, vol.435, pp.239-252, 1990.

[5] W.Rankl, "Smart Card Handbook" 2ed, John Wiley & Sons, 1999.

[6] Jurgensen, Guthery, "Smart Cards The Develo-per's, Toolkit", PHPTR, 1999

[7] 박명수, 김성훈, 김재희, "온라인 서명 검증에서 특징 집합에 대한 각 서명별 가중치 설정 방법", 전자정보통신 논문집 제2권 제1호, pp.62-70, 1995.

[8] 임영이, 이윤철, 강회일, 이동일, "스마트카드 시스템의 보안 기술", 전자통신동향분석 제14권 제5호, pp.42-54, 1999년 10월

[9] 주학수, 현진수, 성재철, 임선각, "IC카드의 안전성 관련 기능 및 공격기법", 정보보호학회지 제13권 제4호, pp.88- 101, 2003년 8월

[10] Cheol-han Park, Dae-wha Seo, "A Design of Expandable IC Card Operating System", 통신정 보보호학회 논문지 제9권, 제2호, 1999.

[11] 윤석창, "스마트카드를 이용한 키 분배방식에 관한 연구", 세명논총 제6집, pp.257-266

[12] Smart Cards and Security Overview, http://www.smartcardbasic.com

[13] 이장원, 홍기융, 조현숙, "스마트카드를 이용한 네트워크 가입자 신분 확인", 한국정보처리학회 논문지 제3권 제5호, pp.1170-1178, 1996.

[14] GEMPLUS, GPK4 Reference Manual, GEMPLUS, 1999.

[15] 이민섭, "현대 암호학", 교우사

[16] CHAN, Siu-cheung Charles, "An Overview of Smart Card Security"

[17] 이경호, 차영태, 심주걸, 원동호, "직접적 인증을 제공하는 안전하고 효율적인 키동의 프로토콜", 한국정보처리학회 논문지, 제6권 제12호, pp.3613-3621, 1999.

[18] A. Frier, P. Karlton, and P. Kocher, "The SSL 3.0 Protocol", Netscape Communications Corp, Nov 18, 1996.

[19] 김진형, "온라인 서명 검증의 현황 및 방법론 소개", KAIST, 2001

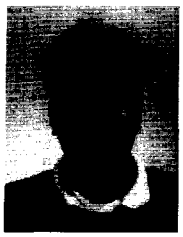
[20] 원지연 외4, "IC 카드를 이용한 생체인식 기술 개발 동향", 한국전자통신연구원.

[21] SoftForum, "암호와 보안 프로토콜", http://www.softforum.com

[22] D. Pinkas and R. Housley, "Delegated Path Vali-dation and Delegated Path Discovery Protocol Requirements", RFC 3379, February 2001.

[23] A. Sorkin, "LUCIFER, A Cryptographic Algori-thm", Cryptologia, Vol.8, No.1, pp.22-24, 1973.

저 자 소 개



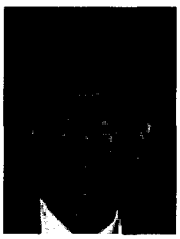
송 영 상(종신회원)

1998년 삼척산업대학교 전자공학과 학사

2000년 단국대학교 전자컴퓨터공학과 석사

2004년~현재 단국대학교 전자컴퓨터공학과 박사 과정

<주관심분야: 정보보안, 전자상거래, 스마트카드, 자바카드>



신 인 철(종신회원)

1973년 고려대학교 전자공학과 학사

1978년 고려대학교 전자공학과 석사

1986년 고려대학교 전자공학과 박사

<주관심분야: 병렬처리, 정보보안, 스마트카드, 자바카드, 디지털 워터마킹, 전자상거래>