다양한 응용을 위한 스마트카드 운영체제
(Smart Card Operating System for Various Applications)

김 증 섭 † 조 병 호 † 김 효 철 †† 이 종 국 ††† 유 기 영 ††††
(Jeung-Seop Kim) (Byoung-Ho Cho) (Hyo-Cheol Kim) (Jong-Kook Lee) (Ki-Young Yoo)

요 약 본 논문에서는 다양한 응용을 위한 스마트카드 운영체제 시스템의 설계 및 구현 방법에 관해 기술한다. 스마트카드는 마이크로프로세서와 메모리를 가지는 독립적인 시스템인데, 전자 상거래와 전자 화폐 등의 높은 보안성을 요구하는 다양한 응용에 사용될 수 있다. 스마트카드 운영체제는 스마트카드 기동의 베이스를 제공하고, 응용 프로그램을 제어하고 감독하는 역할을 수행한다. 스마트카드 운영체제가 EEPROM에 다양한 응용을 지원하기 위한 파일시스템을 생성하여 제어하여야 하고, 외부 장치와의 명령어 및 메시지를 통신할 수 있어야 하고, 명령어의 처리 및 응답 메시지를 생성할 수 있어야 하고, EEPROM에 파일 보안과 통신상의 보안 기능을 제공하여야 한다. 따라서, 본 논문에서는 카드와 터미널간의 인증(authentication), 다양한 응용을 위한 세션(session) 인증, 명령어 처리, 보안성 유지 등의 기능을 수행하는 스마트카드 운영체제 시스템을 설계하고 구현한다.

키워드 : 스마트카드, 스마트카드 운영체제(scos), 스마트카드 보안성

Abstract In this paper, we describe a design and implementation method of a smart card operating system for multi applications. A smart card is the independent computing system and is able to be used in multi applications such as the electronic commerce and the electronic cash. Smart card operation system(SCOS) provides a basis of smart card booting, and controls and manages application programs. SCOS can produce and control a file system to support multi applications in EEPROM, communicate commands and messages with outside devices, process a command, produce a reply message, and provide security functions of file security in EEPROM, and communication security. Therefor, in this paper, we design and implement SCOS system that provides the authentication between a card and a terminal, the session authentication for multi applications, the processing of the commands, and the maintenance of the security.

Key words : smart card, smart card operating system(SCOS), smart card security

1. 서 론

스마트카드는 기존의 플라스틱 카드에 마이크로프로세서와 메모리 등을 내장한 IC 칩과, 8개의 접속단자를 통하여 외부의 카드 리더기로부터 전원 및 데이타 송수신을 하는 독립된 연산 장치이다. 이는 각별한 보안을 필요로 하는 전자상거래와 전자 화폐, 전자 지갑, 전자 주민등록증, 의료 카드 등의 다양한 응용 분야에서 사용될 수 있다. 그리고, 스마트카드 운영체제가 카드 내부의 ROM에 상주하여 카드와 카드 리더기간의 인증(authentication), 명령어 처리, 명령어 처리시의 보안 유지 등의 작업을 수행한다. 이러한 작업을 수행하기 위해서, 스마트카드 운영체제는 카드와 카드 리더기와의 통신, 비휘발성 메모리에 데이타 쓰기, 읽기, 지우기 등의 기본적인 기능과 보안 유지를 위한 암호화 기능을 수행하여야 한다[1, 2].

현재 사용되는 신용카드, 의료카드, 주민등록증 등 대부분의 카드는 하나의 목적으로만 사용하고, 개인 신용정보가 카드에서 쉽게 읽혀지고, 제삼자의 개입이나 보안성 문제에 대해서 제약점을 가지고 있다. 그러므로,

† 비 회 원 : 경북대학교 컴퓨터공학과
damibi@dittotec.com
bhcho@dittotec.com
†† 비 회 원 : 계명문화대학 컴퓨터정보계열 교수
khc@ns.km-c.ac.kr
††† 비 회 원 : 한국전자통신연구원 정보보호연구본부 연구원
kookiss@dreamwiz.com
†††† 종신회원 : 경북대학교 컴퓨터공학과 교수
yook@knu.ac.kr
논문접수 : 2001년 8월 20일
심사완료 : 2002년 3월 13일

차세대 스마트카드는 다양한 응용 분야를 하나의 카드에서 서비스할 수 있어야 하고, 개인의 비밀 유지 등의 보안성 문제를 해결할 수 있어야 한다. 스마트카드는 내부 EEPROM에 파일 시스템을 구축하는데, 멀티 세션을 지원할 수 있어야 하고, 각각의 세션은 카드에서 제공되는 각각의 응용 서비스를 담당하여 사용자 데이타 파일 및 해당 세션의 비밀키 파일을 저장한다. 그리고, 스마트카드에서의 보안 유지는 스마트카드와 외부 터미널 사이에 통신상의 보안과 EEPROM의 각각의 파일 접근에 관한 보안 유지 기능이 제공되어야 한다. 이러한 다양한 응용을 위한 멀티 세션과 보안 기능은 차세대 스마트카드 운영체제에서 제공되어야 하는 필수적인 기능이다[3, 4, 5, 6].

본 논문에서는 다양한 응용을 위한 멀티 세션을 지원하고 보안 기능을 가지는 스마트카드 운영체제를 설계 및 구현한다. 스마트카드 운영체제는 파일시스템, 보안 시스템, 명령어 실행 및 입출력(I/O)시스템의 기능을 가져야 하고, 본 논문에서 이러한 기능을 수행하는 스마트카드 운영체제를 설계하고 구현한다. 스마트카드 내부의 파일은 MF(master file), DF(dedicated file), EF(elementary file)로 구성되는데, 스마트카드 파일시스템은 EEPROM에 MF, DF, EF의 파일을 트리(tree) 구조로 설계한다. 스마트카드 운영체제에서 제공되는 보안 기능은 통신 선로상의 보안과 파일 접근에 대한 보안으로 구분된다. 통신 선로상의 보안 기능은 보안 메시징(secure messaging) 기법으로 안전한 메시지 송수신이 되도록 설계한다[5, 7]. 보안 메시징 기법은 메시지를 암호화를 해서 송신하고, 수신할 때 해독하여 원문을 복원하는 기법이다. 암호화 알고리즘은 대칭/비대칭(symmetric/asymmetric) 키 암호화 알고리즘을 사용한다. 그리고, EEPROM에 각각의 파일에 대한 보안 유지는 모든 파일 헤드에 접근 조건(access condition) 필드를 두어 해당 파일의 생성, 읽기, 쓰기, 삭제 등의 파일 접근을 제어할 수 있게 설계한다. 명령어 실행 및 입출력은 스마트카드에서 수신한 명령어를 실행하기 위해서 하드웨어 요소인 EEPROM 및 입출력 단자를 접근하기 위한 명령어 처리와 오류 발생시 오류 메시지를 생성할 수 있게 설계한다.

서론에 이어서 2장에서 스마트카드 시스템 구조, 메모리 구조 및 스마트카드 명령어 구조에 대한 국제표준인 ISO/IEC 7816-1,2,3,4를 관련 연구로서 소개하고, 3장에서 다양한 응용을 위한 스마트카드 파일 시스템을 설명하고, 4장에서 스마트카드에서 제공되는 보안 기능을 설명하고, 5장에서 스마트카드 명령어 처리에 대해서 설명하고, 6장에서 실험 결과를 보인 후에, 7장에서 결론을 맺는다.

2. 관련 연구

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 입출력 장치(write/read unit WRU)간의 전기 신호 특성과 프로토콜 등을 규정하고, ISO/IEC 7816-4에서는 카드와 WRU간의 통신에 필요한 데이타 구조, 카드내의 파일 구성, 보안 체계(security architecture) 등을 규정하고 있다[1, 7, 8, 9, 10, 11]. 다음 절에서부터 국제 표준인 ISO/IEC 7816-1,2,3,4에서 정의한 스마트카드 시스템 구조와 메모리 구조 및 명령어 구조를 설명한다.

2.1 스마트카드 시스템 구조

스마트카드는 신용카드 크기의 플라스틱 카드에 0.3mm 두께의 마이크로프로세서와 메모리를 가진 IC 칩을 내장한 카드를 말한다. 스마트카드의 물리적인 구조와 스마트카드에 내장된 IC 칩의 내부 구조는 ISO/IEC 표준 IC 카드의 형태를 따르고, 그림 1과 같다. 자기 띠(magnetic stripe) 영역과 양각(emboss) 영역은 일반적인 플라스틱 카드와 같은 영역에 배치되어 있다.

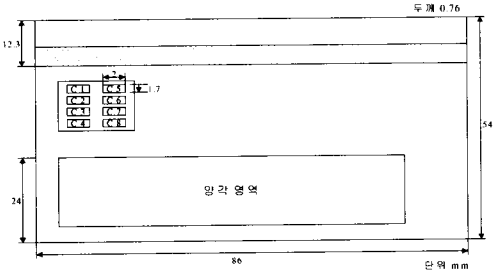


그림 1 스마트 카드의 외부 구조

스마트카드는 8개의 접속 단자를 가지고 있고, 카드의 앞 또는 뒤에 위치할 수 있다. 8개의 접속 단자들 중에서 사용되는 접속 단자는 6개이고, 2개는 미래에 사용할 수 있도록 준비(reserved future use, RFU)해 둔 것이다. C1은 전원 공급부(power supply), C2는 리셋(reset), C3은 클락(clock), C5는 접지(ground), C6은 EEPROM을 지우거나 프로그래밍, C7은 입출력(I/O)의

용도로 사용된다.

스마트카드 내부는 정보를 처리하는 마이크로프로세서, 정보를 저장하는 메모리, 정보의 입출력을 담당하는 입출력 장치로 구성된다. 스마트카드의 마이크로프로세서는 주로 보조 처리기(coprocessor) 없이 8비트가 대부분이지만, 차세대에는 보조처리기를 가진 마이크로프로세서 또는 32비트 마이크로프로세서 형태로 발전될 것이다. 마이크로프로세서의 기능은 ROM에 이식된 운영체제 프로그램을 통하여 정보를 처리, 저장, 입출력 장치로의 이동 등을 관리한다. 스마트카드 시스템의 메모리는 일반적으로 16Kbytes의 ROM, 16Kbytes의 EEPROM, 1 Kbytes 미만의 RAM으로 구성된다. 스마트카드와 WRU 간의 통신은 접속 단자들을 통하여 이루어진다. 스마트카드 시스템 구조는 그림 2와 같다.

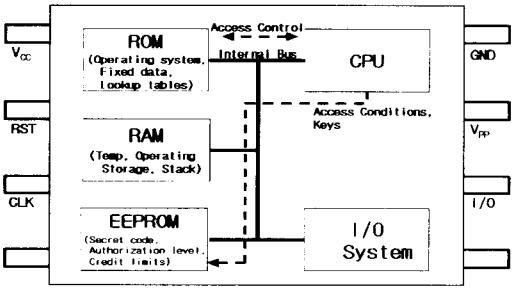


그림 2 스마트카드 시스템 구조

2.2 스마트카드 메모리 구조

스마트카드 시스템에서 메모리 구조는 그림 3과 같이 각각의 기능과 시스템 주소를 가진다. ROM은 스마트카드 운영체제의 기계어 코드가 상주하는데, 본 논문에서 구현한 스마트카드 운영체제의 크기는 약 10Kbytes (2800h)이고, 나머지 6Kbytes는 예비 영역으로 설계하였다. EEPROM은 파일 시스템이 상주하는 곳으로, 192 bytes의 시스템 영역, 32bytes의 MF, 32bytes의 Issuer EF, 80 bytes의 키 EF 및 일반 DF와 EF를 위한 사용자 파일 영역으로 설계하였다. RAM은 1Kbyte 정도의 크기로, 운영체제가 사용하는 스택이나 버퍼를 위한 영역으로 설계하였다.

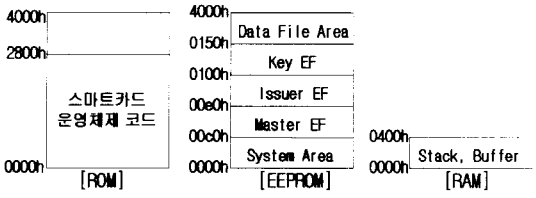


그림 3 스마트카드 메모리 구조

2.3 스마트카드 명령어 구조

스마트카드 운영체제는 명령어와 응답 메시지를 사용하여 WRU와 스마트카드 사이에 통신을 한다. WRU가 명령어를 송신하면, 스마트카드는 명령어를 수신하여 처리하고, 그 결과인 응답 메시지를 WRU에게 송신한다. 스마트카드와 터미널 사이에서 송수신되는 명령어와 응답 메시지는 APDU(application protocol data unit)의 구조를 따르고, 그림 4와 같다[11].

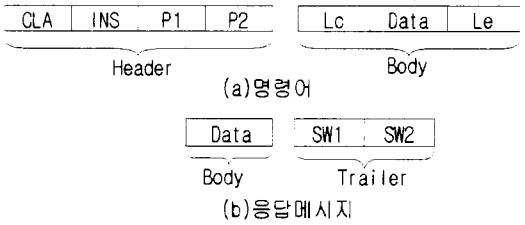


그림 4 명령어와 응답메시지 구조

CLA 필드는 크기가 한 바이트이고, CLA필드의 값이 0x8X이면 ISO 세션에서 사용되고, CLA필드의 값이 0xX8이면 송신되는 명령어가 암호문이고, 0xX0이면 송신되는 명령어는 평문이다. INS필드는 크기가 한 바이트이고, 명령어 구분 코드이다. P1, P2필드는 크기가 한 바이트씩이고, 명령어에서 사용되는 변수 1과 변수 2이다. Lc필드는 크기가 한 바이트 또는 세 바이트이고, 송신하는 데이타의 바이트 수를 나타낸다. Le필드는 크기가 세 바이트 이하이고, 수신하는 데이타의 바이트 수를 나타낸다. Data 필드는 크기가 Lc 또는 Le필드의 값에 해당되는 바이트를 가지고, 송신 또는 수신하는 데이타를 나타낸다. SW1, SW2 필드는 크기가 한 바이트씩이고, 응답 메시지에서 사용하는 상황 표시 1과 상황 표시 2이고, 정상적인 명령어 처리인 경우에는 0x9000의 값

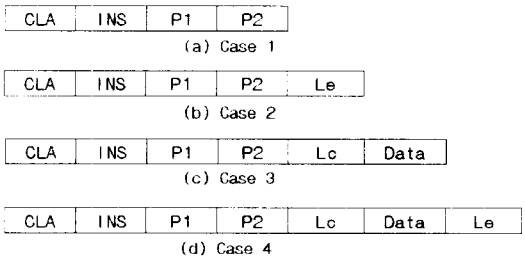


그림 5 APDU 명령어 구조

표 1 APDU 명령어의 구분

구분	명령어 데이타	응답 메시지 데이타
Case 1	No Data	No Data
Case 2	No Data	Data
Case 3	Data	No Data
Case 4	Data	Data

을 가진다.

스마트카드와 WRU 사이에 송수신되는 명령어와 응답 메시지의 APDU(application protocol data unit) 구조는 명령어 데이타와 응답 메시지의 데이타의 존재 유무를 근거로 네 종류로 나눌 수가 있고, APDU 명령어 구조는 그림 5와 같고, 명령어의 구분은 표 1과 같다[11].

3. 스마트카드 파일 시스템

스마트카드 파일 시스템은 EEPROM에 생성되는 MF, DF, EF의 계층적인 구조와 각각의 파일 구조를 관리하고 접근 제어하는 시스템이다. 스마트카드의 파일들은 MF를 루트(root)로 하여 트리(tree) 구조의 형태로 계층을 이루고 있는데, MF 밑에는 DF와 EF 모두 존재할 수 있고, DF 밑에 DF가 존재하는 멀티 레벨 구조와 DF 밑에 EF들만이 존재하는 2레벨 구조가 있다.

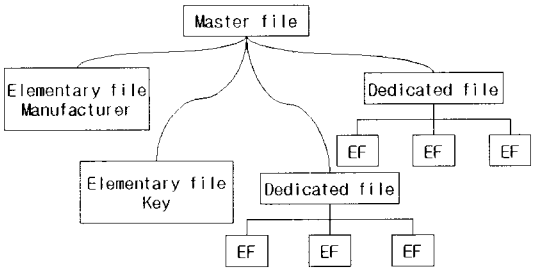


그림 6 스마트카드 파일 시스템의 계층 구조

본 논문의 스마트카드 파일 시스템은 그림 6과 같이 2레벨 구조로 설계하였다. 하나의 응용이 하나의 DF에 해당하여 좀 더 쉽게 파일 접근을 할 수 있게 설계하였고, 복잡한 응용의 경우에는 상위 DF의 다음 파일 주소를 하위 DF의 시작 주소로 하여 멀티 레벨 구조로 쉽게 확장할 수 있게 설계하였다. 스마트카드의 모든 파일은 16바이트의 파일 헤드와 몸체로 구성되며, 파일 헤드는 10개의 필드를 가지고 있는데, 각각의 필드가 파일의 기능적 의미를 가지고 있고, 몸체는 EF만이 가질 수 있도록 설계하였다.

3.1 스마트카드 파일 시스템 구조

본 논문에서는 스마트카드 운영체제가 MF, DF, EF의 생성에서 접근 및 관리에 관한 작업을 수행하도록 스마트카드 파일시스템의 구조를 설계하였다. EEPROM에 시스템 영역은 스마트카드 운영체제가 필요로 하는 데이타와 파일 시스템의 파일 관리 및 제어하는데 필요로 하는 데이타를 가지고 있다. 표 2는 본 논문에서 구현한 EEPROM 안에 시스템 영역에 있는 데이타를 하드웨어 주소와 함께 보여주고 있다.

표 2 시스템 영역의 메모리 맵

주 소	내 용
0000h	Chip Serial Number
0008h	부팅 DF(MF)의 주소 및 ID
000Ch	LOCK_1(세션 선택에 관한 lock byte)
000Dh	LOCK_2(DF 또는 EF에 관한 lock byte)
000Eh	EEPROM의 마지막 주소
0010h	이용 가능한 EEPROM의 시작 주소
0012h	TSK(transport secret key)
0016h	ATR 주소
0018h	RFU(reserved future use)
0038h	ATR의 바이트 수
0039h	ATR
0040h	MF 주소
0042h	DF 주소
00C0h	Master File의 시작 주소

파일의 위치는 레벨(level)로 나타낼 수 있는데, MF는 레벨 0이 되고, MF 밑에 있는 DF나 EF들은 레벨 1이 되고, DF 밑에 있는 모든 EF들은 레벨 2가 된다. 레벨은 파일 헤드의 접근 조건에서 비밀키 파일의 위치를 찾는데 이용된다. 모든 파일은 파일 헤드에 2 바이트의 파일 ID를 가지고 있는데 MF는 항상 0x3F00이고, DF는 파일 이름에 의해서 참조될 수 있다.

EF에는 키 파일, 응용 제어 파일, 작업 파일 등이 있고, 키 파일과 응용 제어 파일은 키 또는 파일 헤드의 접근 조건 필드에 의해 보호되어 있고, 키 파일은 비밀키를 저장하고, 응용 제어 파일은 응용 프로그램의 제어 정보를 저장하고, 작업 파일은 응용 데이타를 저장한다. EF의 구조는 정적 선형 구조(linear fixed), 가변 선형 구조(linear variable), 순환 구조(cyclic), 순차적인 구조(transparent)가 있고, EF 내의 데이타를 참조하는 방법

에는 레코드 참조(record reference)와 바이트 참조(byte reference)가 있는데 본 논문의 모든 파일은 순차적인 구조이고, 일반 파일은 바이트 참조를 하고, 키 파일과 특정 응용의 파일들은 레코드 참조를 한다.

스마트카드 파일 시스템은 EEPROM에 생성되는 MF, DF, EF의 계층적인 구조와 MF에서 각각의 DF와 EF를 접근하기 위해서 연결 리스트(Linked list)를 이용한다. 연결 리스트를 통해서 스마트카드의 모든 파일을 접근하기 위해서, 모든 파일 헤드에 Next Address 필드를 이용하여 각각의 파일을 접근한다. 그림 7은 Next Address 필드를 이용하여 파일 접근을 보여주고 있다.

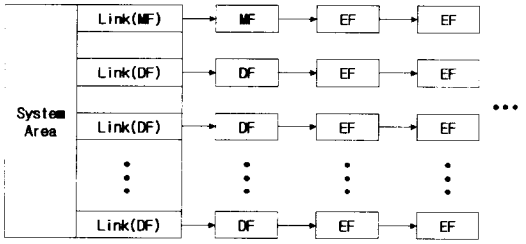


그림 7 Next Address 필드를 이용한 파일 계층

이러한 연결 리스트를 이용한 파일 계층은 스마트카드에서 파일의 생성 및 삭제를 빠르고 쉽게 할 수 있는 장점을 가지고 있고, 파일의 크기가 정적인 경우에는 메모리 단편화 없이 항상 EEPROM의 상태를 최적화할 수 있게 된다. 그러나, 파일의 크기가 동적인 경우에는 스마트카드의 EEPROM에 메모리 단편화 현상이 발생하게 된다.

3.2 스마트카드 파일 구조

스마트카드의 파일시스템에서 MF, DF, EF는 16바이트의 파일 헤드와 몸체를 가지고 있고, 파일 헤드는 10개의 필드를 가지고 있다. 각각의 필드가 파일의 기능적 의미를 가지고 있다. 디렉토리를 나타내는 MF와 DF의 몸체는 단지 이름 또는 식별자를 저장하고 있고, EF의 몸체는 사용자 데이타를 저장하고 있다. 표 3은 MF와 DF의 헤드를 보여주고 있고, 표 4는 EF의 헤드를 보여주고 있다.

표 3 Master File과 Dedicated File의 헤드 구조

Next address		File ID	
File No	Descriptor	Body size	
AC_1		AC_2	
AC_3		Status	Checksum

표 4 Elementary File의 헤드 구조.

Next address		File ID	
File No	Descriptor	Body size	
AC_1		AC_2	
AC_3		RFU	Checksum

Next address 필드는 크기가 두 바이트이고, 파일시스템에서 링크(link) 필드의 의미를 가진다. 파일이 생성될 때 초기 값은 0x0000이다. 예를 들어, DF를 생성하고 EF 2개를 DF 밑에 생성해보자. DF를 생성하면, DF의 Next address 필드는 0x0000이다. 그리고, 첫 번째 EF를 생성하면, 첫 번째 EF의 주소가 DF의 Next address 필드에 저장되고, 첫 번째 EF의 Next address는 0x0000이 된다. 두 번째 EF를 생성하면, 두 번째 EF의 주소가 첫 번째 EF의 Next address 필드에 저장되고, 두 번째 EF의 Next address 필드는 0x0000이 된다. 이상과 같은 방법으로 DF 밑의 모든 EF들은 모두 접근할 수 있다. MF와 레벨 1의 모든 EF들도 이와 같은 방법으로 생성하면 레벨 1의 모든 EF들을 접근할 수 있다. 그리고, MF와 모든 DF들의 주소는 EEPROM의 시스템 사용영역에 저장되어 있다.

File ID 필드는 크기가 두 바이트이고, 해당 파일의 인식자(identifier)로 사용된다. File ID 필드의 하위 다섯 비트는 짧은 ID(short identifier)로 사용하기도 한다. File NO 필드는 크기가 한 바이트이고, 같은 레벨의 파일들 중에 생성된 순서의 의미를 가진다. Descriptor 필드는 크기가 한 바이트이고, 해당 파일의 종류를 알 수 있고, Descriptor 필드의 구조는 그림 8에서 보여주고 있고, 설명은 표 5와 같다.

7	6	5	4	3	2	1	0
0	0	X	X	X	0	0	0

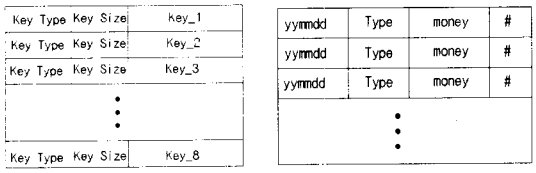
그림 8 파일 데스크립터(Descriptor) 구조

표 5 파일 데스크립터(descriptor)의 구분

Bit 5	Bit 4	Bit 3	파일 종류
1	1	1	Dedicated File
1	0	0	Key Elementary File
0	1	1	Application Elementary File
0	0	0	All other Elementary File

Body Size 필드는 크기가 두 바이트이고, 해당 파일에서 헤드의 크기를 제외한 몸체의 크기를 나타낸다. AC_1, AC_2, AC_3 필드는 모두 크기가 두 바이트씩이고, 해당 파일의 접근 조건을 나타낸다. 접근 조건 필드에 대해서는 4절 스마트카드의 보안 기능에서 자세히 설명한다. DF 헤드의 Status 필드는 크기가 한 바이트이고, DF에만 존재하는 필드이고 해당 DF 밑에 키 파일을 가지고 있는지 여부를 나타낸다. Checksum 필드는 크기가 한 바이트이고, 파일 헤드에서 Checksum 필드와 다른 모든 필드들의 배타적 OR(exclusive OR) 연산을 해서 0x00의 값을 가지게 한다.

EF의 몸체는 EF의 종류에 따라서 조금씩 다르다. EF의 종류는 작업 파일, 키 파일, 응용 파일 등이 있다. 작업 파일은 데이타를 순차적으로 바이트 구조로 저장한다. 키 파일, 응용.파일 등은 데이타를 레코드 구조로 저장한다. 키 파일, 응용 파일의 몸체 구조는 그림 9와 같다.



(a) 키 파일 몸체 (b) 응용 파일 몸체

그림 9 레코드 구조의 파일 몸체

키 파일 몸체에서 Key Type필드는 ISO 세션과 응용 세션 중에 어느 세션에서 사용되는지 여부를 나타내는데, 0x00이면 ISO 세션에서 사용되고, 0x01이면 응용 세션에서 사용된다. 응용 파일 몸체에서 Type필드 값에 따라서 의미가 구분된다. #필드는 다음 레코드와 구분을 위해 분리 첨자로 사용된다.

4. 스마트카드 보안 시스템

본 논문의 스마트카드 보안 시스템은 두 가지 형태의 보안 기능을 제공한다. 스마트카드와 외부 WRU간의 통신 선로상의 보안 기능과 스마트카드 EEPROM에 있는 모든 파일 접근에 관한 보안 기능이다. 전자의 보안 기능은 통신 선로상의 데이타를 보호하기 위해서 보내는 쪽에서 암호화를 수행하고, 받는 쪽에서 복호화를 수행한다. 여기에서 사용되는 암호 알고리즘은 대칭키 암호화 알고리즘과 비대칭키 암호화 알고리즘이 사용된다. 후자의 보안 기능은 파일 헤드에 있는 접근 조건 필드를 이용하여 해당 파일의 접근을 제어한다.

4.1 통신 선로상의 보안

스마트카드와 외부 WRU간의 통신 선로상의 보안 기능은 보안 메시징 기법(Secure Messaging)을 이용하여 카드와 WRU간의 양방향 인증과 사용자 인증 및 카드의 세션별 암호화의 세 가지 기능을 제공한다. 보안 메시징 기법은 데이타를 송신할 때 암호화를 해서 송신하고, 수신할 때 복호화를 해서 데이타를 수신하는 기법이다. 통신 선로상의 보안은 두 가지 성질을 만족해야 하는데, 첫째는 노출 방지이고, 둘째는 변형 방지인데, 본 논문의 시스템은 대칭키 암호화 알고리즘과 비대칭키 암호화 알고리즘을 이용하여 노출 방지와 변형 방지에 뛰어난 보안 성능을 보여 준다. 대칭키 암호화 알고리즘은 Two Key Triple-DES 알고리즘을 사용하고 식(1)과 식(2)와 같이 암호 및 복호를 수행하고, 비대칭키 암호화 알고리즘은 RSA 알고리즘을 사용하고, 식(3), 식(4)과 같이 암호 및 복호를 수행한다. 비대칭키 암호화 알고리즘은 스마트카드와 WRU간의 양방향 인증, 사용자 인증, 세션 인증 및 카드와 터미널 사이에 키를 공유하는데 사용되고, 대칭키 암호화 알고리즘은 송수신되는 데이타를 암호 또는 복호하는데 사용된다.

$$E(M) = E_{K1}(D_{K2}(E_{K1}(M))) = C \quad \text{(식 1)}$$

$$D(C) = D_{K1}(E_{K2}(D_{K1}(C))) = M \quad \text{(식 2)}$$

$$E(M) = M^e \bmod N = C \quad \text{(식 3)}$$

$$D(C) = C^d \bmod N = M \quad \text{(식 4)}$$

M은 평문이고, C는 암호문, 식(1)과 식(2)에서 K1, K2는 56 비트 DES 키이고, 식(3)과 식(4)에서 공개키는 (e, N)이고, 비밀키는 (d, N)이다.

스마트카드와 WRU간의 양방향 인증은 그림 10과 같은 과정으로 수행된다.

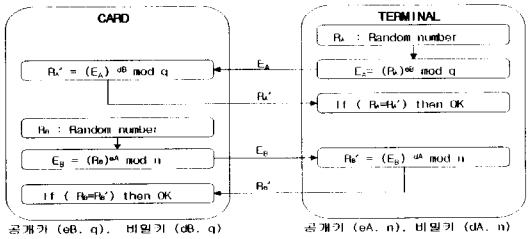


그림 10 스마트카드와 WRU간의 양방향 인증

1) WRU는 난수 R_A를 생성하여, 카드의 공개키 (eB, q)를 이용하여 RSA 암호화를 한 E_A를 스마트카드로 송신한다.

2) 스마트카드는 수신한 E_A를 자신의 비밀키 (dB, q)를 이용하여 복호화를 한 R_A'를 WRU에게 송신한다. 이와 동시에 카드에서 난수 R_B를 생성하여, WRU의 공개키 (eA, n)를 이용하여 RSA 암호화를 한 E_B를 WRU에게 송신한다.

3) WRU는 R_A와 R_A'을 비교해서 같으면 WRU가 스마트카드를 인증한다. 그리고, 자신의 비밀키 (dA, n)를 이용하여 RSA 복호화를 한 R_B'을 스마트카드에게 송신한다.

4) 스마트카드는 R_B와 R_B'을 비교해서 같으면 스마트카드와 WRU를 인증한다.

스마트카드와 WRU는 인증을 할 때마다 새로운 난수를 상대방의 공개키를 이용해서 암호화를 하고, 복호화를 할 경우에는 자신의 비밀키를 이용해서 복호화를 하기 때문에 보안성이 다른 암호 알고리즘에 비해 높다. 그리고, 통신 선로 상에서 해커의 도청이 있다고 하더라도 암호화된 수 또는 난수만이 송수신되고 있으므로, 해커의 공격을 무산시킬 수 있다.

사용자 인증은 스마트 카드의 주인임을 인증하는 것이고, 다음 그림 11과 같은 과정으로 수행된다.

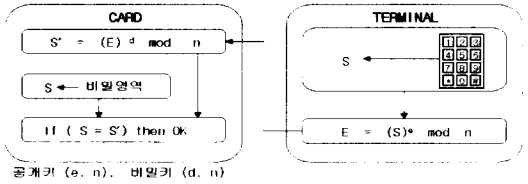


그림 11 사용자 인증

1) 사용자는 WRU에서 사용자 비밀번호 S를 입력한다.

2) WRU는 카드의 공개키인 (e, n)을 이용해서 암호화한 수 E를 스마트카드에게 송신한다.

3) 스마트카드는 자신의 비밀키인 (d, n)을 이용해서 복호화한 수 S'과 카드 내의 비밀영역에 있는 S를 비교해서 같으면 사용자를 인증한다.

세션별 암호화는 스마트카드에서 제공되는 세션에서 카드와 WRU간의 송수신 메시지를 Two-Key Triple DES 알고리즘을 이용하여 메시지를 암호화한다. 스마트카드와 WRU간의 두 개의 56 비트 키의 공유는 세션이 시작될 때, 카드 내의 비밀영역에 있는 두 개의 56 비트 키를 WRU의 공개키인 (eA, n)를 이용하여 RSA 암호화를 해서 송신하고, WRU는 자신의 비밀키인 (dA, n)를 이용하여 RSA 복호화를 해서 카드와 WRU간의 키를 공유한다.

4.2 스마트카드 내의 파일 보안

스마트카드 내의 모든 파일은 세 개의 접근 조건을 파일 헤드에 가지고 있는데, 접근 조건 필드는 두 바이트 크기이고, 파일의 접근 여부에 관계된 키 파일의 다섯 비트 짧은 ID와 키의 순서를 가지고 있다. 해당 파일의 비밀키 파일은 해당 파일의 레벨과 같거나 상위에 존재한다. 그림 12는 접근 조건 필드를 바이트 구조로 보여주고 있다.

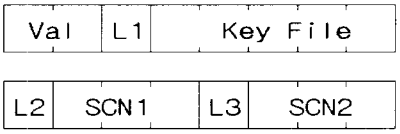


그림 12 접근 조건의 구조

Val속성은 접근 조건의 의미를 나타내는데 표 6에서 그 값의 의미를 정의하고 있다. L1속성은 키 파일의 레벨을 나타내고, Key File속성은 키 파일 헤드의 두 바이트 ID필드에서 하위 다섯 비트를 짧은 ID로 사용한다. L2속성은 첫 번째 비밀키를 가진 파일의 레벨을 나타내고, SCN1속성은 첫 번째 비밀키의 순서를 나타내고, L3속성은 두 번째 비밀키를 가진 파일의 레벨을 나타내고, SCN2속성은 두 번째 비밀키의 순서를 나타낸다.

표 6 접근 조건의 Val 필드 값과 의미

값	의미
00	비밀키로 보호되어 있지 않다.
01	SCN1로 참조되는 비밀키로 보호되어 있다.
10	SCN2로 참조되는 비밀키로 보호되어 있다.
11	접근 절대 불가하다.

DF 파일은 3개의 접근 조건 필드를 가지고 있고, 접근 조건 필드는 각각 2바이트로 구성되어 있다. 첫 번째 접근 조건 필드는 시스템 파일에 관계된 조건으로 시스템 파일의 생성과 삭제여부의 조건을 의미하고, 두 번째 접근 조건 필드는 일반 파일에 관계된 조건으로 일반 파일의 생성과 삭제여부의 조건을 의미하고, 세 번째 접근 조건 필드는 RFU로 사용된다. 시스템 파일에는 MF, 비밀 파일, 키 파일, 그리고 스마트카드 제조에 관계된 파일 등이 있고, 일반 파일에는 시스템 파일을 제외한 모든 작업 파일 등이 있다. DF의 접근은 3회까지

선택 기회를 주고 그 이상의 오류를 발생하면 해당 DF는 블락 상태로 들어가고, 동작 상태로 회복하기 위해서는 해당 세션의 세션키를 이용하여 회복한다.

EF 파일도 3개의 접근 조건 필드를 가지고 있고, 접근 조건 필드는 각각 2바이트로 구성되어 있다. 첫 번째 접근 조건 필드는 읽기 기능에 관계된 조건이고, 두 번째 접근 조건 필드는 쓰기 기능에 관계된 조건이고, 세 번째 접근 조건 필드는 지우기 기능에 관계된 조건이다. EF의 접근도 DF와 마찬가지로 3회까지의 선택 기회를 주고, 그 이상의 오류를 발생하면 해당 EF가 속한 DF는 블락 상태로 들어가고, 동작 상태로 회복하기 위해서는 세션키를 이용한다.

5. 스마트카드 명령어 처리

터미널에서 스마트카드로 송신하는 명령어는 스마트카드의 직렬 통신 포트를 통해서 수신되고, 스마트카드 운영체제가 명령어 처리 루틴을 호출하여 명령어를 실행하고, 그 결과 응답 메시지를 터미널로 송신한다.

명령어 처리 루틴은 그림 13과 같이 명령어 처리 흐름도에 따라서 명령어가 실행된다.

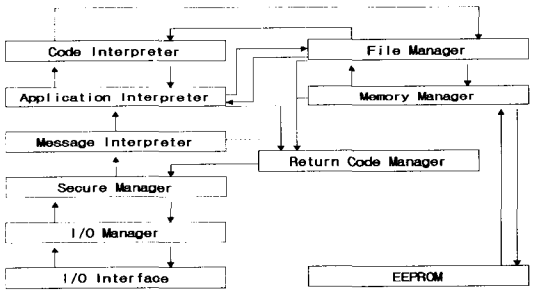


그림 13 명령어 처리 흐름도

명령어 처리 루틴은 9개의 함수적 모듈을 가지고 명령어를 처리한다. I/O Interface는 RS232C 포트를 통해서 비동기 글자 전송(Asynchronous Character Transmission) 프로토콜 방식으로 WRU와 통신한다. I/O Manager는 입출력 명령어를 보내거나 받기 위한 처리를 한다. Secure Manager는 입출력 명령어를 암호화 또는 복호화 기능을 수행한다. Message Interpreter는 입력된 명령어를 파싱(parsing)하는 모듈이다. 스마트카드 명령어는 APDU의 구조로 되어 있지만, 통신을 위해서는 TPDU(Transmission Protocol Data Unit)의 구조로 변경되어야 하고, Message Interpreter에서 TPDU 구조의 명령어를 APDU의 구조의 명령어로 변환한다. Application Interpreter에서는 입력된 명령어가 4가지 종류의 명령어를 구별하여 해당 모듈을 호출한다. Code Interpreter는 명령어 실행을 위한 모듈이다. File Manager는 명령어 실행 과정에서 EEPROM의 메모리를 접근을 관리, 감독하는 모듈로써 DF, EF의 선택에서 키 파일 검사와 접근 조건 필드 검사를 수행한다. Memory Manager는 직접 EEPROM에 데이타를 쓰기, 읽기, 지우기 등의 하드웨어적인 일을 수행한다. EEPROM은 Memory Manager만이 접근할 수 있기 때문에, 스마트카드 사용자(User)나 발급자(Issuer)에게는 투명한 안전성(Transparent Security)를 제공할 수 있다. Return Code Manager는 입력된 명령어의 결과로 SW1, SW2의 상태 바이트를 생성하는데, 명령어 실행이 성공일 때는 0x9000의 값을 보내고, 실패했을 때는 각 해당 모듈에서 발생하는 오류 코드를 보낸다.

스마트카드 운영체제에서 명령어 처리를 그림 13과 같이 모듈화하여 처리함에 의해서 다양한 응용을 위한 명령어 생성이 용이하고, 명령어 처리 과정을 분석과정 명확해지는 장점을 가지고 있다. 스마트카드 명령어 처리 각 모듈은 상호 독립적으로 동작함으로써 운영체제 기능을 확장 및 응용 범위를 확장할 수 있다.

6. 실 험

6.1 실험 환경

다양한 응용을 위한 스마트카드 운영체제를 구현하기 위해서, 스마트카드로는 80196 PCB(printed circuit board)와 8051 PCB를 사용하였다. 80196 PCB와 8051 PCB는 RTOS(real time OS) 개발을 위한 범용적인 PCB이고, 80196 PCB는 인텔(Intel) 계열의 기계어에 동작하고, 8051 PCB는 필립스(Philips) 계열의 기계어에 동작한다. 스마트카드로 구현된 보드의 구조는 80196 마이크로컨트롤러 또는 8051 마이크로컨트롤러, 32 Kbytes의 ROM, 32 Kbytes의 RAM, RS232C 입출력 장치, 리셋 장치, 16 bits 어드레스 버스, 그리고 8 bits 데이타 버스로 구성되고, WRU로 구현된 것은 펜티엄 컴퓨터이다. 스마트카드와 WRU 사이의 통신 속도는 9600 보레이트(baud rate)이다. 스마트카드 운영체제 소스 코드는 C 언어와 어셈블리 언어를 사용하여 작성하였고, C 컴파일러는 80196 PCB의 경우 ic96을, 8015 PCB의 경우 KEIL 컴파일러를 사용하였고, 80196 PCB의 어셈블러는 asm96을 사용하였다. WRU에서는 비쥬얼 프로그램인 Microsoft Visual C++ 버전 5.0을 이용하여 80196 보드와 WRU 사이의 통신 프로그램, 인터페이스 프로그램, 응용 프로그램 등을 구현하였다.

그림 14는 스마트카드인 8051 PCB 보드와 WRU인 PC 및 스마트카드와 WRU 사이의 통신, 응용 인터페이스 프로그램의 실험 환경이다.

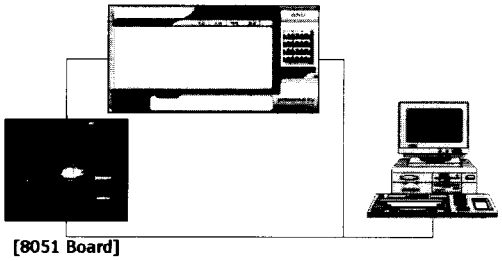


그림 14 스마트카드 시스템의 실험 환경

6.2 스마트카드 운영체제 시스템의 실험

본 실험은 스마트카드 운영체제의 동작에 관한 실험이다. 스마트카드 운영체제의 동작은 ATR(Answer-To-Reset) 단계, 양방향 인증 단계, 응용 서비스 단계의 세 단계로 구성되고, 그림 15와 같다.

ATR 단계는 스마트카드에 전원을 공급하거나 초기화를 수행하였을 때, 스마트카드가 WRU에게 ATR 메시지를 송신하는 단계이다. 전원이 처음 공급될 때는 콜드 리셋(Cold Reset)이라 하고, 초기화를 수행할 때는 웜 리셋(Warm Reset)이라 한다. ATR 메시지는 최대 33 글자로 구성될 수 있고, 스마트카드가 WRU에게 전압, 전류, 클럭 사이클(Clock Cycle), 사용 가능한 프로토콜 등의 전기적, 물리적 환경 요소를 전송하는 단계이다. WRU는 ATR 메시지를 해석한 이후, 스마트카드와의 통신을 시작할 수 있게 된다.

양방향 인증 단계는 WRU가 ATR 메시지를 받은 이후, 무작위 수 RN을 RSA 암호화 함수(f())를 통해서 암호화된 메시지를 스마트카드로 보내고, 스마트카드는 RSA 복호화 함수(g())를 통해서 RN'를 복호화한다. 그리고, 스마트카드에서 무작위 수 rn을 RSA 암호화 함수(f())를 통해서 암호화된 메시지를 생성하여, RN'과 함께 WRU에게 보낸다. WRU는 RN과 RN'이 같으면 WRU 입장에서 스마트카드를 신뢰하게 되고, RSA 복호화 함수(g())를 통해서 rn'을 복호화해서 다시 스마트카드로 보낸다. 스마트카드에서 rn과 rn'이 같으면 스마트카드 입장에서 WRU를 신뢰하게 된다.

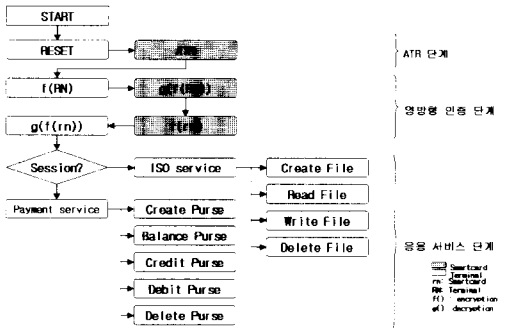
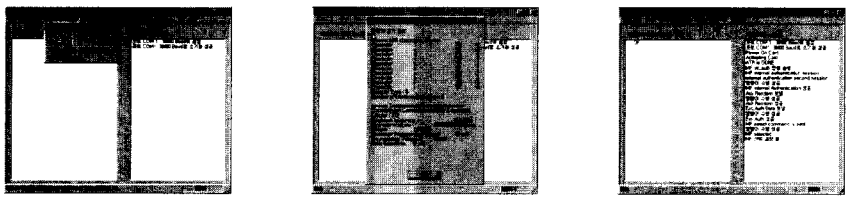


그림 15 다양한 응용을 위한 스마트카드 운영체제 흐름도

응용 서비스 단계는 세션 인증을 통해서 어떤 응용을 수행할 것인지를 결정하고, 해당 응용 서비스의 명령어를 수행할 수 있게 된다. 세션에서 작업을 완료하고, 다른 세션을 선택할 때는 다시 세션 인증을 받아야만 서비스를 받을 수 있게 된다.

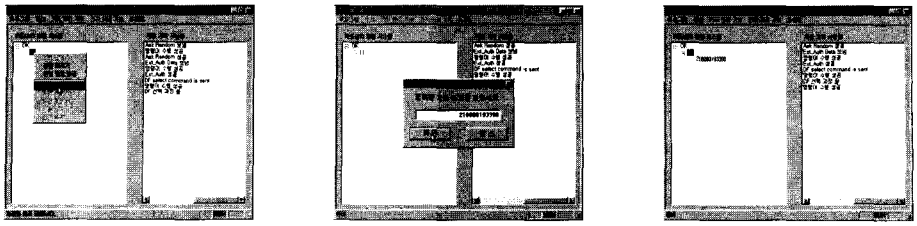
본 실험에서는 지불 세션을 그 예로 구현하였다. ATR 단계는 그림 16에서와 같이 a)카드 접속, b)ATR 메시지 확인, c)MF 선택의 순서로 동작한다. 스마트카드에서 WRU로 보내는 ATR 메시지는 카드와 WRU간의 전류, 전압, 프로토콜 등에 대한 정보를 동기화하는 데에 목적을 둔다.

스마트카드에서 전자 지갑 생성은 그림 17에서와 같이 a)전자 지갑 생성 선택, b)전자 지갑 이름 작성, c)전자 지갑 생성 완료의 순서로 동작한다. 각 그림에서 오른쪽 화면은 스마트카드 내부의 파일시스템 구조가 트리 구조로 생성됨을 알 수 있다.



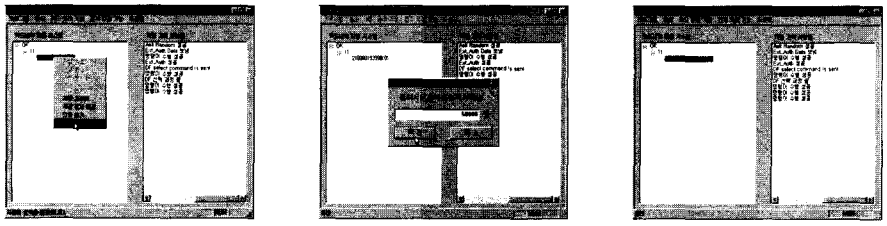
a) 카드 접속 b) ATR 메시지 확인 c) MF 선택

그림 16 ATR 단계



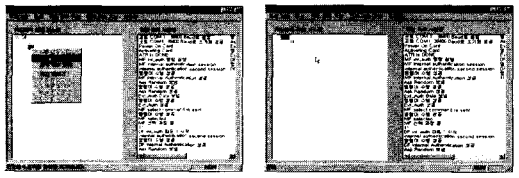
a) 전자 지갑 생성 선택 b) 전자 지갑 이름 작성 c) 전자 지갑 생성 완료

그림 17 전자 지갑 생성



a) 전자 지갑 입금 선택 b) 입금 금액 입력 c) 전자 지갑 입금 완료

그림 18 전자 지갑 입금



a) 전자 지갑 삭제 선택 b) 전자 지갑 삭제 완료

그림 19 전자 지갑 삭제

그림 17과 같이 스마트카드 EEPROM에 전자 지갑을 생성하고, 그림 18의 전자 화폐 입금과 그림 19의 전자 지갑 삭제를 통하여 실제로 스마트카드가 전자 화폐의 기능을 수행함을 살펴보자. 이와 같은 기능은 스마트카드 내부에 파일에 대한 생성, 읽기, 쓰기, 지우기의 기본적인 파일 처리 기능을 구현한 것이다.

본 실험에서 구현한 스마트카드는 전자 지갑뿐만 아니라 의료 카드, 신용 카드, 신분 카드 등의 용도로 사용 범위를 확장할 수 있고, 처리 내용 및 데이타를 스마트카드 내부 메모리에 저장하고 있기 때문에 높은 보안 기능을 가지고 있다. 현재, 우리 나라는 K-Cash 전자 화폐를 표준으로 사용하고 있고, K-Cash 전자 화폐의 프로토콜이 국제 표준인 ISO/IEC 7816을 따르고 있기 때문에, ISO/IEC 7816을 준수하는 본 논문의 스마트카드에 쉽게 이식할 수 있다.

스마트카드의 대부분의 응용이 스마트카드 내부 메모리에 파일 생성, 읽기, 쓰기, 지우기 등의 기본적인 기능을 수행하고 있기 때문에, 본 논문의 스마트카드는 다양한 응용을 쉽게 구현할 수 있고, 다양한 응용을 위한 세션 인증 기능을 가지고 있기 때문에, 세션별 파일 관리, 사용자 관리, 키 관리 및 보안 기능 등을 제공한다. 또한, 본 논문의 스마트카드 운영체제는 대칭/비대칭 키를 이용한 암호화 기능을 수행하고 있기 때문에 높은 보안 기능을 가지고 있다.

본 실험에서 구현한 스마트카드 명령어는 국제 표준인 ISO/IEC 7816-4에 있는 명령을 수행하는 ISO 세션 명령어와 전자 지불 명령을 수행하는 지불 세션 명령어로 나눌 수 있고, 그 종류는 표 7과 같다.

ISO 세션에서 사용하는 명령어들의 종류별 기능에서, Internal Authentication 명령어는 ISO 세션을 선택했을 때 사용자의 인증 여부를 확인한다. ISO_DF Create File 명령어는 MF 밑에 ISO DF를 생성한다. ISO_EF Create File 명령어는 ISO DF 밑에 ISO EF를 생성한다. Select File 명령어는 EEPROM에 존재하는 ISO 관련 모든 DF와 EF를 윈도우 운영체제의 탐색기처럼 트리 형태로 볼 수 있게 한다. Read File 명령어는 ISO EF를 바이트 형태로 참조해서 보여 준다. Write File 명령어는 ISO EF에 데이타를 바이트 단위로 첨가한다. Delete File 명령어는 ISO EF를 EEPROM 상에서 지

표 7 세션별 명령어의 종류

종류	명령어
ISO 세션 명령어	· Internal Authentication · ISO_DF Create File · ISO_EF Create File · Select File · Read File · Write File · Delete File · Get Response
지불 세션 명령어	· Payment Session Authentication · Payment_DF Create Purse · Payment_EF Create Purse · Select Purse · Balance Purse · Credit Purse · Debit Purse · Delete Purse · Get Payment Status Response

운다. Get Response 명령어는 스마트카드의 현재 상태 바이트를 WRU로 전송한다.

Payment 세션에서 사용하는 명령어들의 종류별 기능에서, Payment_DF Create Purse 명령어는 MF 밑에 Payment DF를 생성한다. Payment_EF Create Purse 명령어는 Payment DF 밑에 Payment EF를 생성한다. Select Purse 명령어는 EEPROM에 존재하는 Payment 관련 모든 DF와 EF를 윈도우 운영체제의 탐색기처럼 트리 형태로 볼 수 있게 한다. Balance Purse 명령어는 Payment Purse를 레코드 형태로 참조해서 보여 준다. Credit Purse 명령어는 Payment Purse에 데이타를 레코드 단위로 첨가한다. Debit Purse 명령어는 Payment Purse에 데이타를 레코드 단위로 첨가한다. Delete File 명령어는 Payment Purse를 EEPROM 상에서 지운다. Get Payment Status Response 명령어는 스마트카드의 현재 상태 바이트를 WRU로 전송한다.

6.3 실험의 결과 및 분석

본 논문은 다양한 응용을 위한 스마트카드 운영체제를 설계 및 구현하였다. 3장에서 파일시스템을, 4장에서 보안시스템을, 5장에서 스마트카드 명령어 처리 모듈을 설계 및 구현하였다. 파일시스템은 스마트카드 내부 EEPROM에 파일의 생성, 읽기, 쓰기, 삭제의 기능을 쉽게 처리할 수 있게 연결 리스트를 사용하여 파일시스템을 설계하였다. 따라서 MF, DF, EF의 파일들이 체계적으로 생성될 수 있고, 관리될 수 있고, 응용의 복잡도에 따라서 확장 가능한 장점을 가지고 있다.

보안시스템은 통신선로상의 보안과 스마트카드 파일 접근에 관한 보안으로 구분할 수 있는데, 통신선로상의 보안은 대칭/비대칭 키 기법으로 통신 선로상의 어떤 메시지를 도청하더라도 암호문 또는 무작위 수이기 때문에 도청에 대한 위험을 해결하였다고 할 수 있다. 기존의 스마트카드 운영체제에서는 대칭키 암호화 방식을 사용하고 있는데, 대칭키 암호화 방식은 키의 누출에 대한 위험을 가지고 있기 때문에 본 논문에서는 비대칭키 암호화 방식을 통한 대칭키의 공유를 하고 있다. 비대칭키 암호화 방식은 키의 크기에 대해서 암호화 시간이 지수 멱승 시간을 요구하고 있어서, 본 논문에서는 키의 크기를 64 비트의 샘플을 이용하고 있다. 차세대 스마트카드의 CPU가 32비트로 된다면 256 비트 또는 512 비트의 키를 이용한 암호화가 실용화될 수 있다. 스마트카드 파일 접근에 대한 보안은 모든 파일 헤드에 접근 조건 필드를 두어 파일 접근에 필요한 키 값이나 키 파일의 대한 검증을 통해 파일 접근을 허용하고 있기 때문에 모든 파일 내의 데이타를 보호할 수 있다.

스마트카드 명령어 처리 모듈은 명령어의 입력에서부터 응답 메시지의 출력까지 체계적으로 모듈화하여, 각 모듈들이 독립적으로 동작하고 있기 때문에 다양한 응용을 위한 명령어 생성 및 확장이 용이하고, 명령어 처리 분석 과정이 명확해지는 장점을 가지고 있다. 스마트카드 명령어 처리 각 모듈은 상호 독립적으로 동작함으로써 운영체제 기능을 확장 및 응용 범위를 확장할 수 있다.

우리 나라 전자 화폐 표준인 K-Cash 전자 화폐는 국제 표준인 ISO/IEC 7816을 따르고 있다. 본 논문에서 구현한 스마트카드는 ISO/IEC 7816을 준수하기 때문에, K-Cash 전자 화폐를 쉽게 이식할 수 있다. 그리고, 대부분의 스마트카드 응용들이 스마트카드 내부 메모리에 파일 생성, 읽기, 쓰기, 지우기 등의 기본적인 하드웨어적인 기능을 수행하고 있기 때문에, 본 논문의 스마트카드는 다양한 응용을 쉽게 구현할 수 있다.

7. 결 론

본 논문에서는 국제 표준인 ISO/IEC 7816-1,2,3,4를 준수하고, 다양한 응용을 위한 스마트카드 운영체제 시스템을 설계 및 구현하였다. 스마트카드 운영체제는 파일시스템, 보안시스템, 명령어 실행 및 입출력(I/O)시스템의 기능을 가져야 하고, 본 논문에서 이러한 기능을 수행하는 스마트카드 운영체제를 설계하고 구현하였다.

스마트카드 파일 시스템은 스마트카드 메모리에 MF, DF, EF의 파일을 계층적으로 생성되고, 접근 및 메모리 관리 등의 기능을 수행한다. MF, DF, EF를 계층적으로 생성 및 접근하기 위해서 트리 연결리스트를 이용하였다. 스마트카드 보안 시스템은 통신 선로상의 보안

과 스마트카드 내부 파일의 보안 기능을 수행한다. 통신 선로상의 보안은 보안 메시징 기법으로 보안 기능을 하고, 스마트카드 내부 파일의 보안은 접근 조건 필드와 키 파일에 의해서 보안 기능을 한다. 스마트카드 명령어 실행 및 입출력 시스템은 스마트카드와 WRU 사이에 RS232C 통신을 통하여 명령어와 응답 메시지의 통신 기능을 수행하고, 명령어 처리 루틴에 따라서 명령어 입출력에서부터 명령어 해석, 명령어 실행, 결과 메시지 작성까지의 함수적 기능을 수행한다.

스마트카드의 응용으로 전자 지불 시스템을 구현하였는데, 전자 지불 시스템은 스마트카드에 전자 지갑의 생성, 입금, 출금, 조회, 삭제의 기능을 수행하는 전자 지불 명령어를 생성하였다. 스마트카드에서 제공되는 전자 지불 시스템은 전자상거래와 전자 화폐, 전자 지갑 등의 기능으로 사용될 수 있다.

본 논문의 결과로 구현된 스마트카드 운영체제 시스템을 여러 응용분야에 활용하는 방안과 기대되는 성과는 다음과 같다. 첫째, 보안성을 중요시하는 전자상거래, 전자 화폐, 전자 지갑 등의 분야에서 금융 보안의 수단으로서의 활용을 기대할 수 있다. 둘째, 개인 사생활 보호를 필요로 하는 전자주민카드, 의료카드, 운전면허증 등의 개인 신분 확인 및 보호의 수단으로서의 활용을 기대할 수 있다. 이러한 여러 가지 응용을 위해서는 접촉식 및 비접촉식 스마트카드를 통합한 콤비 형태의 스마트카드의 운영체제가 필요하고, 스마트카드 사용자 인증을 위한 방법을 생체 정보를 이용하는 생체 인식형 스마트카드 운영체제 개발에 관한 연구가 필요하다.

참 고 문 헌

[1] J. L. Zoreda, J. M. Oton, *Smart Cards*, ARTECH HOUSE Boston, London, 1994.

[2] 김증섭, "RSA 암호화 기능을 가지는 스마트카드 운영체제 구현", 석사학위 논문, 경북대학교, 1998.

[3] *MPCOS Reference Manual*, GEMPLUS, 1994.

[4] 박철한, "확장성과 적은 메모리 사용을 위한 IC 카드 운영체제의 설계", 석사학위 논문, 경북대학교, 1997.

[5] Chung-Huang Yang, "On the Design of Campus-Wide Multi-Purpose Smart card Systems," Proceedings of the Institute of Electrical and Electronics Engineers 33rd Annual 1999 International Carnahan Conference on Security Technology, IEEE(sp), pp465-468, 1999.

[6] Kathrin Schier, "Multi-functional Smart cards for Electronic Commerce - Application of the Role and Task Based Security Model," *Proceedings of* the Fourteenth Annual Computer Security Applications Conference, IEEE Computer Society, pp147-154, 1998.

[7] W. Rankl, W. Effing, *Smart Card Handbook*, Chanterelle Translations, London, UK, 1997.

[8] ISO/IEC 7816-1, *Identification cards-Inte grated circuit(s) cards with contact-Part 1: Physical characteristics*, 1987.

[9] ISO/IEC 7816-2, *Identification cards-Inte grated circuit(s) cards with contact-Part 2: Dimensions and location of the contacts*, 1988.

[10] ISO/IEC 7816-3, *Identification cards-Inte grated circuit(s) cards with contact-Part 3: Electronic signals and transmission protocols*, 1992.

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

김 증 섭
1997년 경북대학교 컴퓨터공학과 학사.
1999년 경북대학교 컴퓨터공학과 석사.
1999년 ~ 현재 경북대학교 컴퓨터공학과 박사과정. 관심분야는 스마트카드, 지문인식, 정보 보안, 암호학

조 병 호
1995년 경북대학교 컴퓨터공학과 학사.
1997년 경북대학교 컴퓨터공학과 석사.
1997년 ~ 현재 경북대학교 컴퓨터공학과 박사과정. 관심분야는 멀티미디어 시스템, 지문인식, 분산 시스템, 암호학

김 효 철
정보과학회논문지 : 컴퓨팅의 실제
제 8 권 제 1 호 참조

이 종 국
2000년 경북대학교 컴퓨터공학과 학사.
2002년 경북대학교 컴퓨터공학과 석사.
2002년 ~ 현재 ETRI 정보보호연구본부 네트워크 보안 연구부 근무중. 관심분야는 스마트카드, 정보 보안, 암호학

유 기 영
정보과학회논문지 : 컴퓨팅의 실제
제 8 권 제 1 호 참조