# RSA 암호 알고리즘을 이용한 스마트카드의 운영체제 구현*


o
김증섭, 장유탁, 김정준**, 김태근**, 유기영
경북대학교 컴퓨터공학과, 한국통신 무선통신연구소**


# Implementation of Smart Card Operating System using RSA Cryptographic Algorithm


Jeung-Seop Kim Yu-Tak Jang Jung-Jun Kim** Tae-Geun Kim** Kee-Young Yoo
Department of Computer Engineering, Kyungpook National University
Wireless Communications Institute. Korea Telecom**



## 요 약

스마트카드 운영체제는 카드와 터미널간의 인증(authentication), 메시지 처리 및 메시지 처리시 비밀성(security) 유지 등의 작업을 수행한다 본 논문은 스마트카드에서 DES 암호 알고리즘보다 보안성이 뛰어나고, 다양한 응용을 지원하기 위해서 RSA 암호 알고리즘을 이용한 확장 가능한 운영체제를 구현한다. 스마트카드 시스템과 운영체제의 구조는 ISO/IEC 7816 규정을 따르고 있고, 몽고메리 알고리즘을 이용한 RSA 암호 알고리즘은 스마트카드에서 인증과 스마트카드 내에서 파일의 보안성, 메시지 보안 명령어를 안전하게 수행한다 본 논문에서 제시한 스마트카드 운영체제는 다양한 응용을 지원하기 위하여 응용 목적에 따라 운영체제와 응용 프로그램을 확장할 수 있게 설계되었다


## 1. 서 론

오늘날 인터넷을 이용한 전자상거래, 전자화폐 등의 서비스가 제공되고 있고, 이러한 서비스들은 암호 알고리즘을 기반으로 제공되고 있다. 전자상거래, 전자화폐 등의 수행과 암호 알고리즘의 정확한 수행을 위해서 마이크로프로세서와 메모리가 내장된 스마트카드가 등장하게 되었다

스마트카드 운영체제는 스마트카드의 ROM에 상주하여 카드와 터미널간의 인증(authentication) 메시지 처리, 메시지 처리시의 보안성(security) 유지 등의 작업을 지원한다. 이러한 작업을 처리하기 위한 기본적인 작업은 카드와 터미널간의 통신, 비휘발성 메모리에 데이터 쓰기, 읽기, 지우기와 암호 알고리즘의 수행으로 나눌 수 있다. 기존의 스마트카드 운영체제에서는 DES와 같은 대칭키 알고리즘을 사용하고 있다 차세대 스마트카드는 다양한 응용(multi application)을 지원해야 하고, 이러한 응용들을 지원할 수 있는 운영체제가 개발되어야 한다

본 논문에서는 몽고메리 알고리즘을 이용한 RSA 알고리즘을 사용하여 암호화 기능을 수행하는 확장 가능한 스마트카드 운영체제를 제안한다. RSA 알고리즘은 DES 알고리즘에 비해 보안성이 높다. RSA 알고리즘의 수행속도를 개선하기 위해 몽고메리 알고리즘을 이용하여 암호화 기능을 수행한다 하드웨어를 제어하는 기본 운영체제 위에 상위 레벨 운영체제를 설계하여 다양한 응용을 지원한다

본 논문의 구성은 다음과 같다 2장에서는 스마트카드 시스템과 운영체제의 구조에 대해서 설명하고, 3장에서는 스마트카드에서 인증 및 서비스 구현에 대해서 설명한다. 마지막으로 4장에서는 결론 및 향후 연구방향에 대해서 기술한다.

## 2. 스마트카드 시스템과 운영체제의 구조

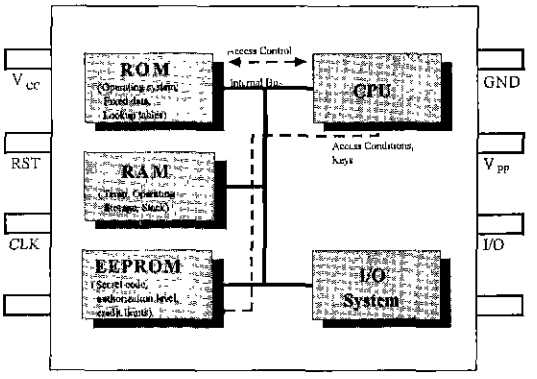
### 2.1 스마트카드 시스템의 구조

일반적으로 스마트카드는 신용카드 크기의 플라스틱 카드에 0.3mm 두께의 마이크로프로세서와 메모리를 내장한 카드를 말한다 ISO/IEC 7816-1[1]에서는 접촉형 IC 카드의 물리적 성격인 카드의 형태, 크기 등을 규정하고 있고, ISO/IEC 7816-2[2]에서는 접점의 크기, 개수, 위치 및 기능 등을 규정하고 있고, ISO/IEC 7816-3[3]에서는 IC 카드와 입출력 장치(write/read unit WRU)간의 전기 신호 특성과 프로토콜 등을 규정하고 있고, ISO/IEC 7816-4[4]에서는 카드와 WRU간의 통신에 필요한 데이터 구조, 카드내의 파일 구성, 보안 체계(security architecture) 등을 규정하고 있다.

스마트카드는 정보를 처리하는 중앙처리장치, 정보를 저장하는 메모리, 정보의 입출력을 담당하는 입출력 장치로 구성된다. 스마트카드의 마이크로프로세서는 ROM에 이식된 운영체제 프로그램을 통하여 정보를 처리, 저장, 입출력 장치로의 이동 등을 관리한다. 스마트카드의 메모리는 RAM(random access memory). ROM(read only


* 본 연구는 한국통신(과제번호 98-13)의 지원에 의한 것임

memory), EEPROM(electrically erasable programmable read only memory)으로 구성된다. ROM에는 스마트카드 운영체제가 저장되고, EEPROM에는 사용자 데이터(application data)가 저장되고, RAM에는 변수들을 저장한다. 접점을 통하여 스마트카드 내부의 데이터가 송수신된다 스마트카드 시스템은 [그림 1]과 같다.

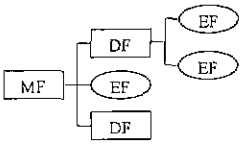



[그림 1] 스마트카드 시스템 구조

구현에 사용된 스마트카드는 80196 마이크로 컨트롤러, 32kbyte의 ROM, 32kbyte의 RAM, RS232C 입출력장치, 리셋장치, 16비트 어드레스 버스, 8비트의 데이터 버스로 구성되고, 통신속도는 9600 보레이트(baud rate)이다. 32kbyte의 RAM 중 16kbyte는 RAM으로 사용하고, 나머지 영역은 EEPROM으로 사용한다. 카드와 통신할 터미널은 일반 PC이고, 터미널에서 제공되는 통신 및 응용 서비스는 C++ Builder 3으로 구현하였다.

2.2 스마트카드 운영체제의 구조

본 논문의 운영체제는 ISO/IEC 7816 규정에 근거한 터미널과의 통신을 통하여 카드의 초기화, 파일 관리, 명령어 처리 및 보안 기능을 수행한다 터미널과의 통신에 사용되는 프로토콜 T0는 비동기 단방향 문자 전송 프로토콜(asynchronous half duplex character transmission protocol)이다 터미널에서 리셋신호가 전송되면 카드는 ATR (answer to reset)을 전송하고, 카드는 ATR을 통해 터미널에게 제공 가능한 프로토콜 및 터미널에게 알려주고자 하는 초기 정보를 전송한다[5, 6, 7].




[그림 2] 파일의 계층 구조

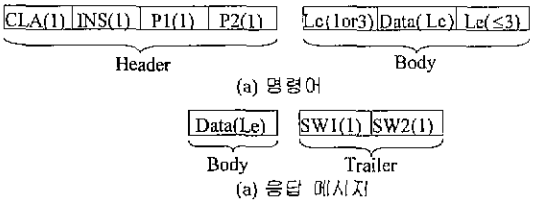
스마트카드 내의 파일은 EEPROM에 위치하게 되고, 외부에서 접근하고자 할 때 접근 조건(access condition) 또는 키를 통해 접근을 제한한다 운영체제가 관리하는 파일에는 MF, DF, EF가 있으며, MF와 DF는 디렉토리이고, EF는 일반 파일이다. 스마트카드 내의 파일의 계층 구조는 [그림 2]와 같다. MF 아래에는 DF와 EF를 만들 수 있으며, DF 아래에는 EF만 만들 수 있는 2 단계 계층구조를 갖는다. EF에는 키 파일과 일반 파일의 2종류가 있는데, 키 파일은 키와 비밀 코드를 저장하고, 일반 파일은 사용자 데이터를 저장한다. 모든 파일은 16 바이트의 헤드와 몸체로 구성되고, 그 구조는 [그림 3]과 같다 MF와 DF의 몸체는 단지 이름 또는 식별자가 저장되고, EF는 데이터를 저장하는 몸체로 구성된다.

| Next address | | File ID |
|---|---|---|
| File No | Descriptor | Body size |
| AC 1 | | AC 2 |
| AC 3 | | Status/RFU \| checksum |

Status : DF, RFU : EF

[그림 3] 파일의 Header

본 논문의 운영체제에서 사용하는 명령어는 2종류로 구분되는데, 관리자 모드의 ISO 명령어와 응용 모드의 Payment 명령어가 있다. ISO 명령어에는 MakeFile, SelectFile, ReadFile, WriteFile, DeleteFile 명령 등이 있고, Payment 명령어에는 CreditPurse, DebitPurse 명령이 있다. MakeFile 명령은 DF나 EF를 생성하고, SelectFile 명령은 파일의 처리에 있어서 먼저 선택을 할 때 사용하고, ReadFile, WriteFile, DeleteFile 명령은 파일에 해당 명령을 수행한다. CreditPurse, DebitPurse 명령은 Purse 파일에 입금과 출금 작업을 수행한다. 명령어는 사용되는 용도에 따라 네가지로 구분되고, 명령어는 헤드와 몸체, 응답 메시지는 몸체와 꼬리로 구성되는데, [그림 4], [표 1]과 같다.




CLA : Command class, INS : Instruction, P1 : Parameter 1, P2 : Parameter 2, Lc : Length of command, Le : Length expected, SW1, SW2 : status byte

[그림 4] 명령어와 응답 메시지 구조

| 구분 | 명령어 데이터 | 응답 데이터 |
|---|---|---|
| 경우 1 | No Data | No Data |
| 경우 2 | No Data | Data |
| 경우 3 | Data | No Data |
| 경우 4 | Data | Data |

[표 1] 4 경우의 명령어

보안이 요구되는 데이터를 전송할 경우, 전송자는 전송되는 데이터를 암호화해서 전송하고, 수신자는 복호화해서 명령어를 수행한다. RSA 암호 알고리즘에서 암호는 식 (1), 복호는 식 (2)를 이용하여 수행한다. 공개키는 (e, N)이고, 비공개키는 (d)이다. 고속 멱승 연산을 수행하기 위해 몽고메리 모듈러 곱셈인 식 (3)이 이용된다.

$$D(C) = C^d \bmod N = M \quad (1)$$

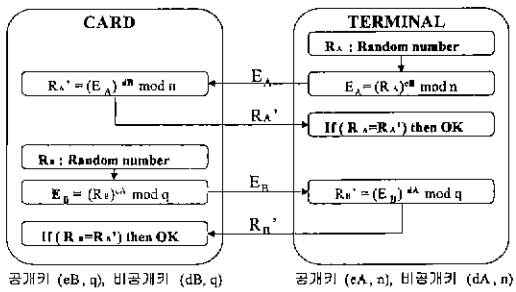
$$E(M) = M^e \bmod N = C \quad (2)$$

$$MMM(A, B, N, R) = ABR^{-1} \bmod N \quad (3)$$

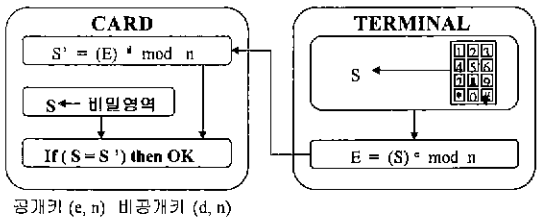
## 3. 인증 및 서비스

3.1 인증

스마트카드를 사용하는 인증에는 터미널인증, 카드인증, 그리고 사용자인증이 있으며[8,9], [그림 5]에서와 같이 터미널인증과 카드인증은 동시에 양방향인증으로 진행되고, 인증할 때마다 난수를 생성하기 때문에 터미널과 카드의 위조방지 효과가 크다. [그림 6]에서와 같이 사용자인증은 응용을 시작할 때마다 진행되고, 사용자가 입력한 패스워드가 스마트카드 내의 비밀영역에 저장된 값과 비교함으로써 이루어 지는데, 패스워드가 터미널에 암호화되지 않은 상태로 노출될 수 있기 때문에 터미널인증이 먼저 수행되어야 한다.




[그림 5] RSA 알고리즘을 이용한 양방향 인증



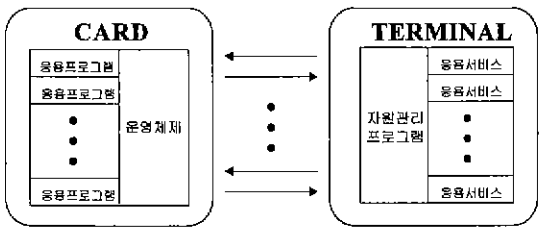

[그림 6] RSA 알고리즘을 이용한 사용자 인증

터미널과 스마트카드와의 인증을 할 때 통신 선로상에 있는 메시지는 암호문 또는 난수이기 때문에, 해커가 통신 선로상에서 도청한 메세지는 복호하기 어렵다.

3.2 서비스

본 논문의 스마트카드에서 지원되는 서비스는 터미널에서 전송한 명령어를 해석하여 명령어를 수행하고, 그 결과를 터미널로 전송하는 기능을 한다. 터미널에서 전송한 명령어는 일반 명령어와 메시지 보안(message securing) 명령어로 구분되고, 응용을 시작할 때 선택한 키를 이용해서 복호한다. 다양한 응용을 지원하기 위해서 터미널에서는 명령어를 응용 서비스별로 구분을 하여 자원 관리자를 통해 특정 서비스의 명령어를 카드로 전송하고, 카드에서는 운영체제가 명령어를 해석하여 관련된 응용 프로그램을 수행한다. 응용 프로그램은 응용 목적에 따라서 운영체제의 함수를 호출하여 명령어를 수행하고, 그 결과를 터미널에 전송한다. 정상적인 수행인 경우는 요구한 데이터가 있으면 데이터와 함께 또는 데이터 없이 상태 바이트 2개를 붙여서 전송하고, 비정상적인 수행인 경우는 상태 바이트에 오류 정보를 붙여서 전송한다.

스마트카드에서 지원되는 응용서비스는 응용의 목적에 따라 확장되고, 운영체제와 응용 프로그램도 확장된다. 스마트카드 운영체제를 [그림 7]처럼 설계함으로써 제조업자는 확장 가능한 운영체제를 구현하고, 발급자는 다양한 응용을 지원하기 위해 운영체제 및 응용 프로그램을 확장하고, 사용자는 하나의 스마트카드를 가지고 다양한 응용으로 사용할 수 있다.




[그림 7] 운영체제와 서비스 구조

## 4. 결 론

본 논문에서는 스마트카드에서 RSA 암호 알고리즘을 이용한 확장 가능한 운영체제를 구현하였다. 스마트카드 시스템과 운영체제 구조는 ISO/IEC 7816 규정을 따르고 있고, 스마트카드를 사용할 때 가장 중요한 문제인 보안사항은 몽고메리 알고리즘을 이용한 RSA 암호 알고리즘으로 터미널 및 카드 인증, 사용자 인증, 명령어를 수행할 때 메시지 보안 등을 수행하고 있다. 그리고, 다양한 응용을 지원하기 위하여 기본 운영체제 위에 상위 레벨 운영체제를 설계하여 응용 프로그램이 이용하게 구현하였다. 향후 연구방향은 ECC(Elliptic Curve Cryptography) 암호 알고리즘, 디지털 서명, 해쉬 등과 JAVA 운영체제를 스마트카드에 적용하려고 한다.

## [참고 문헌]

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

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

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

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

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

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

[7] MPCOS Reference Manual, GEMPLUS, 1994

[8] 스마트카드 시스템 개발 최종보고서, 전자부품종합기술연구소, 1997

[9] 박 철한, "확장성과 적은 메모리 사용을 위한 IC 카드 운영체제의 설계", 석사학위 논문, 경북대학교, 1997