자바 카드를 기반으로 한 UISM 용 T=1 프로토콜의 구현

주홍일* · 한종욱*
*한국전자통신연구원 개인정보보연구팀

Implementation of the T=1 protocol based on Java Card for USIM

Hong-il Ju* · Jong-wook Han*
*Privacy Protection Research Team, ETRI
E-mail : juhong@etri.re.kr

요 약

본 논문은 자바 카드를 기반으로 하는 T=1 프로토콜의 구현에 대해 기술하고자 한다. 본 논문에서 구현된 T=1 프로토콜은 자바 카드 2.2.1의 규격을 기반으로 하며, ISO/IEC 7816 표준을 만족한다. 또한, 구현된 프로토콜의 테스트는 USIM(Universal Subscriber Identity Module) 카드에 적용하여, USIM 카드의 conformance 시험 규격인 3GPP TS 31.122의 전송 프로토콜 시험을 통하여 확인하였으며, 그 결과 규격을 준용함을 확인하였다.

ABSTRACT

This paper describes the design and implementation of the T=1 protocol based on Java Card. The T=1 protocol implemented in this paper complies with ISO/IEC 7816 standard. Also, JCOS(Java Card Operating Systems) including the contactless card protocol conforms to Java Card 2.2.1 specification and is running on 32-bit ARM7TDMI processor. The protocol stack proposed and implemented in this paper is easy to maintenance of protocol independently. To verify the T=1 protocol implemented in this paper we tested the T=1 protocol scenarios defined in ISO/IEC 7816-3 Annex A. And we tested using USIM(Universal Subscriber Identity Module) cards, which include the implemented T=1 protocol. The T=1 protocol was tested and passed all against the specification 3GPP TS 31.122, which was the Conformance Test Specification for USIM cards including the test suites of both transmission protocols.

키워드
자바카드, T=1 protocol, APDU, TPDU

Ⅰ. 서 론

최근 무선 인터넷과 이동 통신의 발전으로 보다 안전한 사용자 인증 및 정보보호 수단으로 프로세서가 내장된 스마트카드의 사용이 점점 늘어나고 있다. 또한, 향후 IMT-2000 환경에서의 안전한 무선인터넷 서비스를 받거나 안전한 전자상거래를 위해서는 스마트카드(USIM : Universal Subscriber Identity Module)를 이용하여 사용자와 단말기간의 상호 인증을 수행하는 기능과 다기능 스마트카드 개발이 요구되고 있다.

본 논문은 이러한 IMT-2000 망의 정보보호 기능을 위해 반드시 필요한 USIM 카드를 개발함에 있어서, 자바 카드를 기반으로 하는 T=1 프로토콜의 구현에 대해 기술하고자 한다. 본 논문은 USIM용 카드에서 T=0, T=1프로토콜 모두를 지원하기 위해 기존의 T=0 프로토콜에 추가로 T=1 프로토콜을 구현하였으며, 또한 기존의 단일 응용 서비스 지원에서 다중 응용 서비스 지원, 보다 안전한 개인 정보의 저장 및 사용, 사용자 인증 및 데이터 인증 등을 포함하는 보다 강화된 보안성 기능을 위해 자바 카드 기술을 기반으로 하는 USIM 카드를 개발하였다.

본 논문의 구성은 1절에서 서론을 간략히 기술하고, 2절에서는 자바 카드 기술, 3절에서는 T=1 프로토콜의 개요, 4절에서는 T=1 프로토콜의 설계 및 구현, 5절에서는 시험 결과에 대해 언급하고, 마지막 6절에서 결론을 기술한다.

II. 자바 카드 기술

일반적으로 자바 카드는 5개의 계층으로 구성되며, 각각 하드웨어 계층, 카드 운영 체제 (Card Operating System) 계층, 자바 카드 가상 기계 (Java Card Virtual Machine) 계층, 자바 카드 API (Application Programming Interface) 계층, 마지막으로 자바 카드 애플릿 계층으로 분류된다. 여기서, COS 위에 탑재된 자바 카드 가상 기계는 플랫폼 독립성을 제공해주며, 그 위에 위치한 자바 카드 API 는 애플릿 프로그램이 참조하는 패키지 형태의 클래스 파일로 애플릿은 프로토콜에 상관없이 같은 APDU(Application Protocol Data Unit) 메소드를 사용 한다. 이러한 특징을 가지는 자바 카드의 구조는 그림 1과 같다[1-3].

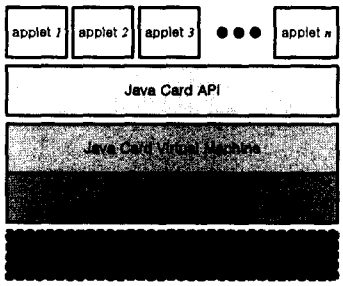

그림 1. 자바 카드의 구조

그림 1과 같은 구조를 가지는 자바 카드의 특징은 자바 카드 가상 기계가 자바 카드 애플릿의 바이트 코드를 수행하고, 메모리나 I/O 같은 스마트 카드 내의 모든 자원에 대한 접근을 제어한다. 또한, 자바 카드는 발급 후에도 최종 사용자가 필요에 따라 다양한 응용 프로그램을 카드에 적재가 가능하다[1-3].

III. T=1 프로토콜의 개요

T=1 프로토콜은 반이중 비동기 블록을 기반으로 하는 전송 프로토콜이며, 프로토콜의 시작은 콜드 리셋(cold reset) 또는 웜 리셋(warm reset)에 의해 카드가 ATR를 보낸 후 또는 PPS(Protocol and Parameters Selection) 프로토콜을 수행한 후 시작된다. 또한, 터미널에서 먼저 블록을 보냄으로써 시작하며, 그에 대한 응답으로 카드가 블록을 보낸다. 이후, 이러한 송수신을 반복함으로써 프로토콜이 수행된다. 블록은 카드와 터미널 사이에 전송되는 가장 작은 데이터 단위이며, 어플리케이션 데이터와 전송 제어 데이터를 포함하여 전송한다. 그리고, 수신된 블록은 에러 체크 후에 전송된 데이터를 처리 한다

카드와 터미널 사이에 전송되는 블록은 프롤로그 부분(prologue field), 정보 부분(information field), 에필로그 부분(epilogue field)로 구성되는데, 프롤로그 부분과 에필로그 부분은 반드시 있어야 하는 필수 사항이며, 정보 부분은 블록의 종류에 따라 선택 사항이다. T=1 프로토콜에서 데이터 전송에 사용되는 블록은 바이트들의 연속이며, 각 바이트는 비동기식 문자로 전송된다. 그림 2는 이러한 T=1 프로토콜의 구조를 보여준다[1][4].

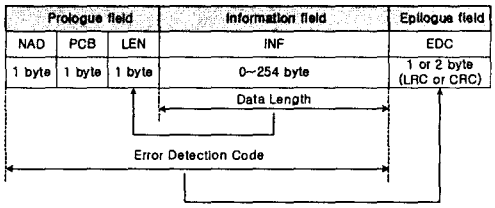

그림 2. T=1 프로토콜의 블록 구조

먼저, 그림 2에서 프롤로그 부분은 NAD(Node address), PCB(Protocol Control Byte), LEN (length) 바이트로 구성된다. NAD 바이트는 블록의 목적 주소(DAD, destination address) 와 소스 주소(SAD, source address) 정보를 포함하고 있는데, 사용하지 않으면, 관련되는 비트는 '0'으로 설정해야 된다. PCB 바이트는 데이터 전송을 제어하기 위해 필요한 정보를 전달하거나 블록의 종류에 대한 정보를 표시하는데, 블록의 종류는 I-블록, R-블록, S-블록으로 구분된다. 여기서, I-블럭은 어플리케이션 층에서 데이터 전송을 위해 사용하고, R-블록은 ACK(Acknowledge) 및 NAK(Not Acknowledge) 정보를 전달하는데 사용하며, S-블록은 카드와 터미널 사이의 제어 정보를 전달하는데 사용한다. 그리고, LEN 바이트는 블록의 정보 부분에 있는 바이트들의 길이를 표시한다. 다음으로 그림 2에서 정보 부분은 I-블럭일 때는 어플리케이션 데이타를 전달하는데 사용되고, S-블럭일 때는 제어 정보를 전달하는데 사용되지만, R-블럭일 때는 사용되지 않는다. 그리고, 블록의 마지막인 에필로그 부분은 블록 전체에 대한 에러 확인 바이트로 사용된다[1][4].

I-블록에서 사용하는 APDU는 카드와 터미널 사이의 어플리케이션 계층에서 사용하는 전송 규약으로 ISO 7816에 정의 되어 있으며, 항상 터미널이 보내는 명령 APDU를 수신하면, 카드 내에서 명령 APDU에 해당하는 작업을 수행하고 응답 APDU를 송신한다. 명령 APDU의 헤더 4 바이트와 응답 APDU의 트레일러 2바이트는 반드시 필요하며, 나머지 바디 부분은 전송되는 데이터 길이에 따라 가변적이다. 명령 APDU는 바디 부분의 Lc, Le에 따라 4가지 경우로 나눌 수 있는데, Lc는 전송하고자 하는 데이터의 길이를 나타내며, Le는 응답을 받고자 하는 데이터의 길이를 나타낸다. 그림 3은 명령/응답 APDU의 구조 및 4가지 경우에 대한 각각의 구조를 보여준다[1][4].

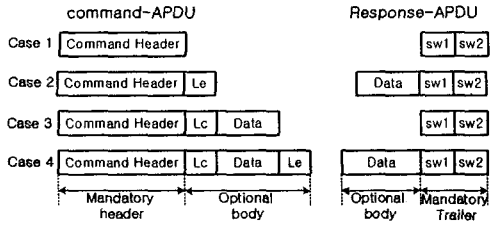

그림 3. 명령/응답 APDU의 4가지 경우에 대한 각각의 구조

그림 3에서 Case 1은 'Lc=0, Le=0' 인 경우, Case 2는 'Lc=0, Le≠0'인 경우, Case 3은 'Lc≠0, Le=0'인 경우, 그리고 Case 4는 'Lc≠0, Le≠0'인 경우를 의미한다. 이러한 APDU는 어플리케이션 층에서 사용되며, 실제 카드와 카드 리더 사이에서는 각 APDU 형식에 해당하는 TPDU(Transport Protocol Data Unit)로 매핑되어 전송된다. 그림 4는 문자 전송 방식인 T=0 프로토콜에서 명령 APDU의 4가지 경우에 매핑되는 각각의 TPDU 수행 과정을 보여주며, 그림 5는 T=1 프로토콜에 대한 경우를 보여준다.

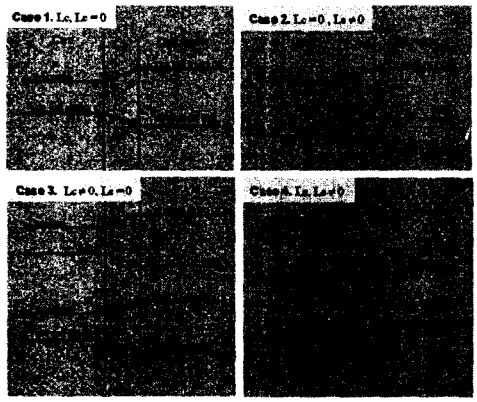

그림 4. T=0 프로토콜에서 명령 APDU의 4가지 경우에 대한 각각의 TPDU 수행과정

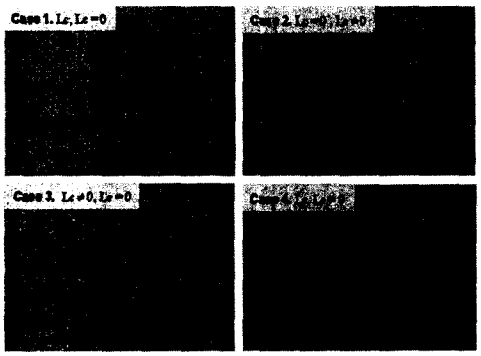

그림 5. T=1 프로토콜에서 명령 APDU의 4가지 경우에 대한 각각의 TPDU 수행과정

IV. T=1 프로토콜의 설계 및 구현

T=1 프로토콜은 물리적 계층, 데이타 연결 계층, 전송 계층, 어플리케이션 계층을 가지는 OSI 모델의 계층적인 구조를 가지는데, 본 논문에서 구현된 T=1 프로토콜은 데이타 연결 계층과 전송 계층에서 C언어로 구현하였다. T=1 프로토콜은 ATR 이후 또는 성공적인 PPS를 수행한후 시작되는데, 그림 6은 프로토콜의 시작 순서도를 보여준다. 카드가 T=0 와 T=1 두 프로토콜을 모두 지원할 경우, 프로토콜의 선택은 PPS 프로토콜에서 포맷 바이트라 불리우는 PPS0 바이트에 의해 결정된다.

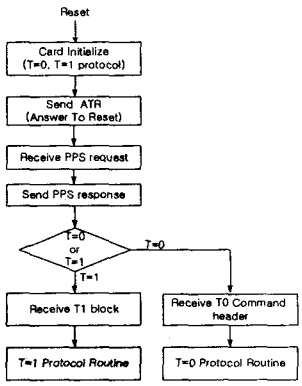

그림 6. 프로토콜의 시작 순서도

그림 6 에서 카드는 터미널로부터 전원과 함께 리셋 신호를 입력 받으면, 카드 초기화가 끝난 후, ATR 데이타를 전송한다. 이때, ATR 신호에는 카드가 지원하는 프로토콜에 대한 정보를 포함하고 있다. ATR을 수신한 터미널은 카드에서 지원하는 T=0 또는 T=1 프로토콜을 선택하기 위해 PPS 요청를 보내고, PPS 응답을 수신한다. 따라서, PPS 교환에서 T=1 프로토콜로 설정되면, 카드는 T1 프로토콜 루틴으로 돌어가게 되며, 이후 송수신하는 데이터 형태는 T=1 블럭의 구조를 가진다.

카드와 터미널 사이의 통신은 터미널의 어플리케이션 계층에서 전송된 명령어 APDU가 카드의 어플리케이션 계층까지 전송되고, 그에 대한 응답 APDU가 카드의 어플리케이션에서 터미널의 어플리케이션까지 전송되는데, 그림 7이 카드와 터미널 사이의 APDU 교환에 대한 블럭도를 보여준다. 그림 7 에서 터미널의 전송 계층은 어플리케이션 계층에서의 명령어 APDU를 아무런 변화 없이 그대로 I-블럭의 정보 부분에 매핑시켜서, 카드의 전송 계층으로 전송한다. 카드의 전송 계층은 수신된 명령어 APDU를 카드의 어플리케이션 계층까지 전달하고, 그에 대한 처리를 수행한 후 전달되는 응답 APDU를 아무런 변화 없이 그대로 I-블럭의 정보 부분에 매핑시켜서, 터미널의 전송 계층으로 전송한다. 이러한 방법으로 통신이 이루어지며, APDU에 대한 TPDU 매핑과 T=1 블럭의 생성 및

전송은 카드와 터미널의 각 전송 계층과 데이타 연결 계층에서 수행하게 된다.

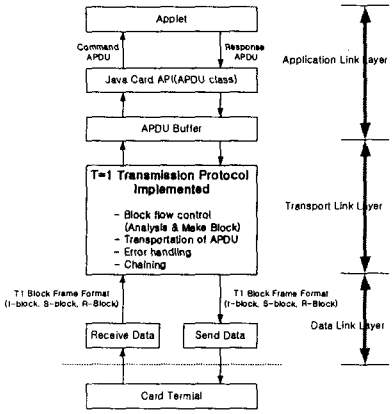

그림 7. 카드와 터미널 사이의 APDU 교환에 대한 블럭도

본 논문에 구현된 T=1 프로토콜은 체이닝(chaining) 기능을 지원하는데, 체이닝 기능은 카드와 터미널 사이의 통신에서 IFSC(Information Field Size for the card) 또는 IFSD(Information Field Size for the interface device)보다 긴 데이터를 전송하고자 할 때, IFSC 또는 IFSD 와 같거나 작게 데이터를 나누어서 통신하는 방법이다. 구현된 T=1 프로토콜의 IFSC 값은 최대 254 바이트까지 지원한다. 또한, T=1 프로토콜을 구현함에 있어서, 자원이 제한된 카드의 메모리 환경을 고려하여 메모리를 효율적으로 사용하도록 구현되었으며, 구현된 T=1 프로토콜을 포함하는 카드는 ATR의 정보 변경이 가능하다. 따라서, 실제 두 프로토콜을 모두 지원하더라도, 둘 중 하나의 프로토콜만 지원하는 카드 또는 두 프로토콜을 모두 지원하는 카드로 ATR 정보를 바꿀 수 있다.

V. 시험 결과

본 논문에서 구현된 T=1 프로토콜의 테스트를 위해서는 ISO/IEC 7816-3 부록-A 에 정의된 T=1 프로토콜 시나리오를 수행함으로써 확인했으며, USIM 카드의 테스트 규격인 3GPP TS 31.122에 대한 테스트를 Aspects 사의 3G 툴을 이용하여 모두 통과하였음을 확인하였다. 3GPP TS 31.122의 테스트는 모든 USIM 카드에 대해서는 필수 사항이며, Aspects 사의 3G 툴에서 USIM Test Suite는 테스트 규격에 대해 규격대로 구현되었는지를 확인 하는 툴이다.

그림 8은 T=1 프로토콜 테스트 중에서 APDU 전송과 관련되는 테스트 결과를 보여주는데, 여기서, 가운데의 녹색 체크는 테스트 통과를 의미하며 오른쪽의 녹색 볼은 성공적인 APDU 통신을 의미한다.

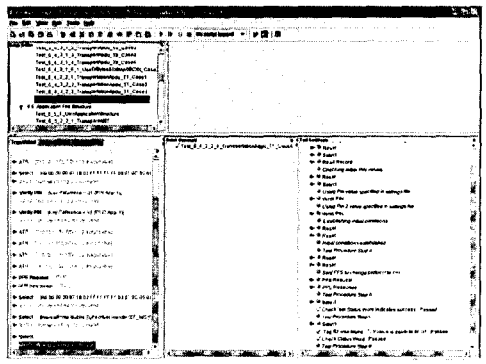

그림 8. USIM conformance 테스트의 결과

VI. 결 론

본 논문은 자바 카드를 기반으로 하는 T=1 프로토콜의 구현 방법을 제안 및 구현하였으며, 이를 USIM 카드에 적용하여 확인하였다. 구현된 T=1 프로토콜은 ISO/IEC 7816 과 자바 카드 2.2.1 규격을 만족하고, USIM conformance 테스트 규격인 3GPP TS 31.122도 모두 통과하였다. 또한, 본 논문에서 구현한 프로토콜 스택은 계층적 접근 방법의 장점을 살려 서로 독립적으로 개발 유지 보수가 용이하게 하였다.

향후 스마트 카드는 점점 여러 분야에서 개인의 정보를 저장하고 보호하는 안전한 매체로 사용될 것이며, 전송되는 데이터의 양도 점점 늘어날 것이다. 특히 이동 통신 분야 등에서 많은 양의 데이터를 빠르게 전송하고자 한다면, 문자 기반의 T=0 프로토콜 보다는 블럭 기반의 T=1 프로토콜이 훨씬 유리하다. 따라서, 향후 다양한 응용서비스를 지원하기 위해서는 하나의 프로토콜 보다는 서비스 종류에 따라 선택할 수 있도록 T=0, T=1 프로토콜을 모두 지원하는 카드가 더 많이 사용될 것이다.

참고문헌

[1] W.Rankl & W.Effing, Smart Card Handbook, John Wiley & Sons, 2000.
[2] Chen, Zhiqun, Java Card Technology for Smart Cards, Addison-wesley, 2000.
[3] http://java.sun.com/products/javacard.
[4] ISO/IEC 7816, International Standards.