스마트 카드를 위한 APDU 패킷 분석기 개발

박진호°, 조증보, 정민수
경남대학교 컴퓨터공학부

A Development of APDU Packet Analyzer Based on Smart Card

ChinHo Park°, JeungBo Cho, MinSoo Jung
Department of Computer Engineering, KyungNam University.

요 약

스마트 카드를 사용함에 있어서 APDU 통신은 필수적이다. 이런 APDU 를 통해 패킷을 주고 받을 경우에 그 패킷의 이진코드를 분석해 자바 가상 기계에 어떻게 할당되는지 시각적으로 보여줌으로써 그 이진코드 하나 하나가 어떤 의미를 내포하고 있는지 보다 명확하고 쉬운 이해가 가능 하도록 하며, 복잡한 절차의 카드와 카드리더기 사이의 APDU 통신을 버튼 하나로 해결해준다.

1. 서론

1.1 연구배경

현재 스마트카드는 다양한 분야에서 활용되고 있다. 이런 다양한 분야에서 스마트 카드를 활용하기 위해서는 각각의 활용분야에 맞도록 작성된 스마트 카드 애플리케이션이 필요하게 된다. 스마트 카드 애플리케이션 개발은 스마트 카드에 장착된 스마트 카드 칩에 대한 지식과 경험이 풍부하지 않은 비전문가들은 불가능 했다. 사실 스마트 카드 운영체제는 여러 문제점을 안고 있다. 다름 아닌 카드 애플리케이션 개발의 어려움과 상호 호환성 부재에 따른 폐쇄적인 환경이다. 이러한 폐쇄적인 환경을 극복하고자 새로운 플랫폼의 요구사항이 대두되었고 그 결과물로써 개방형 플랫폼 개념이 구현되기에 이르렀다. 개방형 플랫폼은 카드가 발행된 후 주어진 카드에 추가될 새로운 애플리케이션 기능을 쉽게 추가할 수 있도록 허용하고 표준언어와 개방형 API 를 사용하여 애플리케이션 개발의 용이성과 개발기간 단축을 도모하고 있다. 개방형 스마트 카드 플랫폼에는 Java Card, MULTOS, Window For Smart Card 등과 같은 세가지의 솔루션이 있다. 이 중 개방형 플랫폼의 선두가 Java Card 이다.

자바로 작성한 파일이 컴파일되어 생성되는 이진코드에는 자바 소스 코드에 나타나 있지 않은 다양한 정보를 내포하고 있다. 본 논문에서 소개하고

있는 스마트카드 패킷 분석기는 카드리더와 카드가 통신을 APDU 통신을 하게 되는데 이때 통신하는 패킷의 형식을 보여주며 그 정보들을 한눈에 볼 수 있다. 먼저 CAP 파일을 APDU 형식으로 변환해서 전송을 하기 위해 SCR 파일을 생성한다. 이러한 SCR 파일에는 많은 정보가 담겨져 있으며 이 SCR 파일의 바이트코드를 분석하고, 그 분석된 결과가 자바 가상 기계 내부구조에 어떻게 할당되는지를 시각적으로 보여줌으로써 패킷이 내포하고 있는 의미를 보다 명확하고 쉬운 이해가 가능하도록 한다. 또한 개발자들에게 좀더 편리한 환경을 제공한다. 그리고, APDU 통신을 할 때 버튼 하나만 누르면 APDU 통신 시뮬레이션을 할 수 있으며 APDU 통신 후 파일 또한 자동으로 생성됨으로써 APDU 통신이 제대로 이루어 졌는지 또한 쉽게 알 수 있으며 스마트 카드의 연구, 개발에 많은 도움이 될 것이다.

1.2 논문의 구성

본 논문의 구성은 다음과 같다. 2 장에서는 자바카드의 전반적인 이해와 On Card / Off Card, APDU 통신에 대해 기술하고 3 장에서는 스마트카드용 패킷 분석기에 대한 설계 및 구현을 다루겠고 4 장에서는 결론 및 향후 동향를 다루도록 하겠다.

2. 관련연구

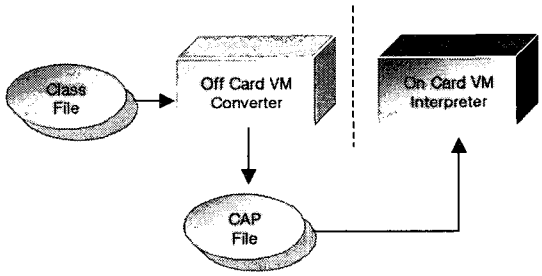
2.1 자바카드의 전반적인 이해(On Card / Off Card)

자바 카드 가상 머신(Java Card Virtual Machine, JCVM)은 스마트 카드를 위해서 JVM 과 다른 구조를 가지고 있다. JCVM 은 일반 자바의 VM(Virtual Machine)과는 달리 스마트 카드에 존재하는 On-Card VM 과 스마트 카드와 연결된 PC 혹은 워크스테이션에 존재하는 Off-Card VM 으로 나누어진다.

이렇게 두 부분으로 JCVM 이 나누어진 이유 중 하나는 보안 문제로 인해 동적 클래스 로딩(dynamic class loading)을 지원하지 않기 때문이기도 하다.

또 다른 이유는 제한된 메모리 문제를 해결하기 위해서 가장 중요한 사항은 스마트카드에서 작동하는 자바 카드 가상 머신의 크기가 작아야 한다는 점이다. 하드웨어 플랫폼에 독립적인 환경을 구축하기 위해서는 자바 카드 가상 머신의 구현이 필수적이며, 이러한 자바 카드 가상 머신은 일반적인 자바 언어 작동을 위한 자바 가상 머신 보다 훨씬 작은 크기를 가져야 한다. 따라서 자바 카드 가상 머신의 구현은 일반적인 자바 가상 머신의 구현과는 다른 모습을 가지게 된다.

이러한 자바카드의 구조는 아래의 [그림 1]으로 이해될 수 있다.



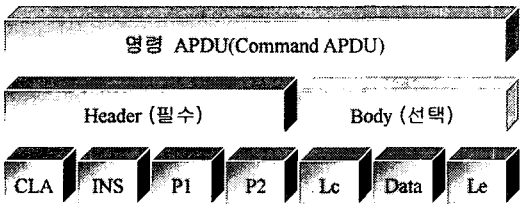
[그림 1] JavaCard Virtual Machine

On-Card VM 은 기본적으로 바이트코드 해석을 담당하고 있는 인터프리터이다. 반면 Off-Card VM 은 클래스 로딩(class loading) 작업에 기반 한 여러 작업을 담당하는 컨버터(converter)이다.

2.2 APDU (Application Protocol Data Unit)

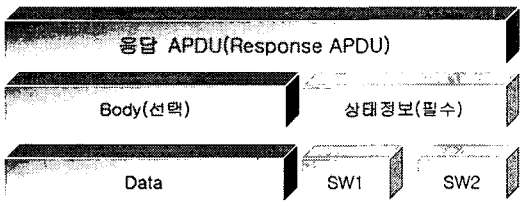
스마트 카드는 호스트 애플리케이션과의 통신 시 APDU(Application Protocol Data Unit)를 사용한다. 이는 ISO7816 에 정의되어 있다. 그러므로 기본적으로 자바 카드 애플릿은 물론 스마트 카드 외부의 애플리케이션 역시 APDU 를 통해 서로 정보를 주고 받는다. 스마트 카드는 호스트 애플리케이션이 주는

명령(Command APDU)를 기준하여 동작하고 이에 반응하여 응답(Response APDU)을 전송한다.
Command APDU 는 기본적으로 헤더와 몸체로 구분하는데 반드시 헤더에 명령어와 해당 명령어의 입력값, 즉 Parameter 를 정의한다. 아래의 [그림 2]는 Command APDU 구조이다.



[그림 2] Command APDU 구조

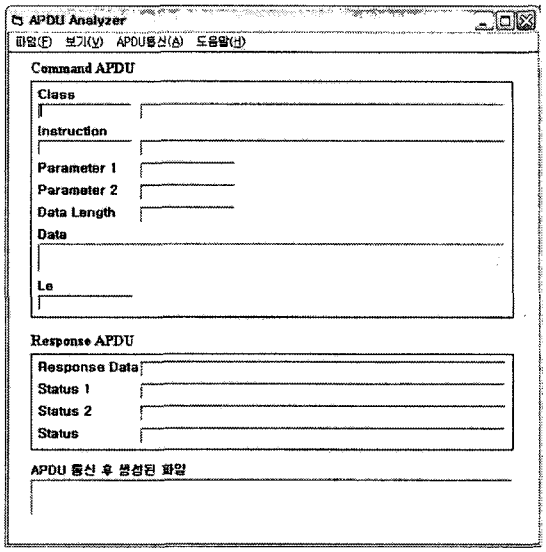
응답 APDU 형식은 매우 단순하다. 스마트카드에 명령이 전달되면 해당 명령어에 대한 데이터를 호스트 애플리케이션에게 응답해야 하며, 데이터는 명령 APDU 에서 정의한 Le 값 만큼 크기를 가진다.
다음의 [그림 3]는 응답 APDU 의 구조이다.



[그림 3] 응답 APDU 의 구조

3. 스마트 카드 패킷 분석기의 설계 및 구현

APDU 통신을 할 때 CAP 파일을 APDU 형식으로 변환해서 전송을 하기 위해 SCR 파일을 생성한다.
이 SCR 파일에는 앞에서 말했듯이 많은 정보가 담겨져 있다. 패킷에 구성요소를 보면 CLA, INS, P1, P2, Lc, Data, Le 가 있다. APDU 통신을 하여 이 패킷이 날아갈 때 이것을 캐치하여 이 패킷의 구성요소들의 이진코드를 보여 줌과 동시에 이 이진코드들이 담고 있는 패킷의 구성 요소를 분석해 주는 것이 목적이다. 다음의 [그림 4]은 스마트카드 패킷 분석기의 GUI 이다.



[그림 4] 스마트 카드 패킷 분석기 GUI

이 분석기 자체에는 버튼 하나만 누르면 APDU 통신을 시뮬레이션 할 수 있는 기능이 내장되어 있다.
APDU 통신을 하고자 할 때 최초는 JCDevelopment Kit 에서 제공하는 scriptgen 명령어를 이용하여 CAP 파일을 입력으로 받아서 스크립트 파일(SCR) 생성한다. CAP 파일은 APDU 형식으로 변환해서 전송하기 위해 SCR 파일을 생성한다. 이런 .SCR 파일을 불러와 Class, Instruction, Parameter 1, Parameter 2, Data Length, Satus 1, Statu 2, Data, Le, Response Data 로 할당되어 분석된다.
APDU 통신 후 .out 확장자(변경가능)의 파일이 자동적으로 생성되게 된다. 이 파일은 APDU통신 시뮬레이션의 결과의 정보를 파일에 내포하고 있으며 GUI 맨 하단에 생성된 파일명을 보여주게 된다. 화면 상단의 파일메뉴는 열기, 저장, 인쇄, 종료 이며 그리고 보기 메뉴는 화면확대, 축소(%로 변경가능) 조절 기능, 그리고 APDU통신에서 시작메뉴는 APDU통신을 해주고 도움말메뉴는 기초적인 지식이 없는 사용자에게 도움을 주고자 만들었다.
이 패킷에 대한 설명은 다음의 [표 1]과 같다.

Class	(1Byte, 필수), CLA(Class Byte)는 카드에 전송하는 명령어 집합을 정의
Instruction	(1Byte, 필수), CLA 에 의해 분류된 명령어는 명령어 바이트(Instruction byte:INS)에서 정의, On Card 파일 시스템에 따라 구조화 될 때 카드의 데이터에 액세스하는 데 사용하는 기본 명령을 지정한다.
Parameter1	(1Byte, 필수), 이 값은 INS 에서 사용하는 입력 값이다. 명령 매개 변수 1 을 정의
Parameter2	(1Byte, 필수), 이 값은 INS 에서 사용하는 입력 값이다. 명령 매개 변수 2 를 정의
Data Length	(전송할 데이터 크기, 선택), Lc(Length of Command data), 명령을 카드에 보낼 때 보낼 데이터의 크기를 의미하며 이 크기만큼 데이터 필드(Data field)에 값을 넣는다.
Data	(Lc 만큼의 Byte), 명령을 실행할 때 필요한 데이터 내용이 포함된다.
Le	(카드가 보낼 응답데이터의 크기), Le(Length of Expected response data), 카드가 명령을 실행하고 보낼 응답 데이터의 크기를 정한다.
Response Data	메소드 실행 시 리턴값이 필요할 때 값이 출력, 값 없을 시 N/A 출력
Status1, 2	(1Byte, 필수) APDU 통신의 성공과 실패에 해당하는 이진코드 출력
Status	Status1, 2 에 해당하는 바이트코드 출력

[표 2-1] 패킷의 구성요소

4. 결론 및 향후 동향

스마트 카드 패킷 분석기를 사용함으로써 스마트카드 분야의 전반적인 지식과 경험이 있는 사람만이 다룰 수 있는 폐쇄적인 환경을 탈피하고자 하는 대목적을 둔다. 동시에 다양한 사람이 쉽게 접근 할 수 있는 환경을 만들어주며, 카드와 카드리더기 사이의 APDU통신 시 날아가는 패킷을 분석함으로써 개발자나 사용자가 손쉽게 그 패킷이 내포하고 있는 코드들의 의미와 바이트코드를 자연스럽게 접하게 될 수 있을 것이다. APDU통신 시뮬레이션 또한 버튼 하나면 자동으로 되어 개발자로 하여금 점진적 개발과 편리한 환경을 제공하여 줄 것이다. 본 논문에서 제시한 스마트 카드 분석기는 자바카드에 국한되어 있지 않을 뿐 아니라, 다양한 응용서비스 개발에 적용하여 활용할 수 있다. 스마트 카드는 과거 사원카드나 캠퍼스카드, 각종회원 카드로 국한되어 사용되어 왔었지만 최근 점차 확대되어 산업분야, 공공분야 금융분야 등 넓은 분야에서 널리 사용될 것으로 바라보고 있다. 세계적인 동향을 보면 북미 지역의 경우 칩 카드 시장이 매우 급속하게 성장할 것으로 예측하고 있으며 전화카드를 중심으로 한 메모리 카드의 퇴조와 스마트 카드의 급성장이 예측된다.

[참고문헌]

[0] Zhiqun Chen, "*Java Card ™Technology for Smart Cards Architecture and Programmer's Guide*", Addison-Wesley, June 2000

[1] Uwe Hansmann, Martin S. Nicklous, Thomas Schack, Frank Seliger, Smart Card Application Development Using Java, Springer, 2002.

[2] Wook-Chol Hwang, "*A Study On the Optimization of the Java Card Virtual Machine Based Smart Card*", Journal of Korea Multimedia Society Vol.4 No.2, 2001

[3] TaeSun Kim, "*Micro Software*", CNETKorea, Inc., February 2004, pp.292-299

[4] DoWoo Kim, "*A Study On The Optimization of Java Class File under Java Card Platform*", Journal of Korea Multimedia Society, December 2003, pp.1200-1207

[5] Sun Microsystems, Inc., *The Java Card ™2.2 Virtual Machine Specification*, SUN, 2002.

[6] Sun Microsystems, Inc., *The java Card ™ 2.2 Runtime Environment(JCRE) Specification*, SUN, 2002.

[7] Bill Venners, "*INSIDE THE JAVA Virtual Machine*", McGraw-Hill, 1998, pp.405-540