Java Card을 이용한 마일리지 통합 관리 시스템 구현

백장미[†] · 강병모[††] · 홍인식[†††]

요 약

인터넷을 통한 전자상거래가 활성화됨에 따라, 사용자에게 편리성을 제공하기 위한 기술이 등장하고 있으며, 보다 편리한 거래를 위하여, 스마트 카드의 활용이 높아지고 있다. 스마트 카드의 차세대 COS로 Java Card가 등장함으로써, 자바 언어를 이용한 다양한 어플리케이션의 개발이 가능해졌다.

본 논문은 차세대 COS로 주목받고 있는 Java Card를 이용하여 인터넷 상에서의 효율성 있는 마일리지 관리 시스템을 제안하였다. 본 시스템은 Java Card상에 저장되는 개인의 독립적인 프로그램으로 Java Card의 연산기능을 이용하여 서로 다른 마일리지 체계를 가지는 쇼핑몰의 마일리지를 직접 계산하고 적립할 수 있다. 기존의 제공되고 있는 마일리지 시스템과의 비교 분석과 구현된 프로그램의 시뮬레이션을 통하여 제안한 시스템의 실효성을 높이고자 한다.

Implementation of Loyalty System using Java Card

Jang-Mi Baek[†], Byung-Mo Kang[††] and In-Sik Hong[†††]

ABSTRACT

As electronic commerce is becoming more popular on the Internet, smart cards have been used for safe transfers and transactions on E-commerce popularly. Especially, Java Card considered as a COS for the next generation must take advantage of the good points of Java Language by using this language and making programs asked for by various demands.

In this paper, we proposed efficient management system of mileage on the Internet using Java Card. The system has security for data and the simplicity of application development by Java Card cryptography. The system is an independent program saved on Java Card and can calculate and save mileage, although the characteristic of the mileage is different from others through the calculating process of the Card. Also, the system is developed to encourage the efficiency of a system after comparing and contrasting between established systems and the newly designed one in simulation.

Key words: Smart Card, Java Card, Mileage System

1. 서 론

인터넷을 통한 전자 상거래가 활성화됨에 따라, 개인의 인증정보나 거래에 관련된 정보 등의 보안성이 강조되고 있다. 이러한 E-commerce 상에서의 안전한 데이터 전송과 거래를 위해 스마트 카드의 이용이 증대되고 있는 추세이다. 특히, 차세대 이동통신에 USIM(Universal Subscriber Identity Module) 카드를 탑재할 예정이다. 스마트 카드는 자체적인 암호와 연산을 통한 고도의 보안성을 제공하고, 데이터의 저장과 연산 기능을 수행하며, 이동성과 개인휴대성이 뛰어나다. 현재 스마트 카드의 시장은 유럽이 선점하고 있으며, 스마트 카드의 대표적인 활용분야는 전화카드와 교통카드를 들 수 있다. 전통적으로 스마트 카드의 제조와 카드의 소프트웨어는 칩 카드 벤더들에 의해 개발되었다. 따라서 칩 의존도가 높았기 때문에 다양한 어플리케이션의 제공과 어플리케이션 사이의 이식성이 존재하지 않았으며, 카드의 개발

본논문은 정보통신부의 지원을 받아 수행되었음.
[†] 준회원, 순천향대학교 대학원 전산학과 석사과정
[††] 준회원, 순천향대학교 대학원 전산학과 박사과정
[†††] 정회원, 순천향대학교 공과대학 정보기술학부 부교수

비용도 높아서 시장 형성 자체의 장기화를 초래하였다. 그러나 Hardware적으로 독립적인 COS(Chip Operating System)가 등장함으로써 독자적인 카드 어플리케이션을 개발할 수 있게 되었다. 어플리케이션의 개발 환경으로는 Open Platform Card로 썬 마이크로 시스템의 Java Card, 마이크로 소프트의 WFSC(Windows for Smart Card), 마스터 카드의 MULTOS 등이 있다. 특히 차세대 COS로 떠오르고 있는 썬 마이크로 시스템의 Java Card는 자바언어를 사용하기 때문에 자바 언어의 특성을 최대한 활용할 수 있으며 다양한 요구의 어플리케이션을 효과적으로 개발할 수 있다[1].

본 논문은 Java Card를 기반으로 하여, Java Card 상에서의 관리가 수월한 마일리지 시스템을 개발하고자 한다. 본 시스템은 USIM 카드에 저장하여 모바일 기기의 사용을 위한 어플리케이션으로 활용이 가능하다. 제안한 시스템은 지불과 관련된 마일리지 통합 관리 시스템으로 Java Card에 저장되는 한 개인의 독립적인 프로그램이다. Java Card의 연산기능을 이용하여 서로 다른 마일리지 체계를 가지는 이종 쇼핑몰간의 통합 마일리지를 직접 계산하고 적립할 수 있다. 기존의 제공되고 있는 마일리지 시스템과의 비교 분석과 구현된 프로그램의 시뮬레이션을 통하여 제안한 시스템의 실효성을 높이고자 한다. 본 시스템의 시뮬레이션 환경은 자바언어를 사용한 J빌더에서 구현되었으며 ISO 7816 스펙을 기준으로 하였다. 시뮬레이션을 위한 기본적인 카드와 단말기는 Java Card COS를 지원하는 Gemplus의 Gemxpress 211을 이용하였다[2,3].

2. 관련기술

2.1 스마트 카드(smart card) 기술

2.1.1 스마트 카드의 정의

스마트 카드란 신용카드와 유사한 형태의 카드로, 마이크로프로세서와 메모리를 내장하고 있어서 정보의 저장과 연산처리기능이 가능하다. Chip 카드나 IC 카드라는 용어로 사용되기도 한다. 스마트 카드는 1968년 독일에서 Jurgen Dethoff, Helmut Grotrupp에 의해 처음 소개되었으며 프랑스나 독일 등의 유럽 지역에서는 전화카드나 금융카드로 활용하고 있다. 스마트 카드가 출현하게 된 가장 큰 원인은 보안 제공 능력이 뛰어나기 때문이다. 보안 블록(security block) 기능을 제공하여 COS에 의해서만 액세스가 가능하게 하며, 디렉토리마다 보안 블록을 두어 디렉토리 내의 파일 접근을 제어한다. 단말기로부터 자신이 정당한 카드 소지자임을 인증하기 위한 단계이며 8bit의 PIN(Personal Identification Number)를 부여한다. 이러한 특징은 인터넷과 전자상거래에서의 스마트 카드 이용을 보다 활발하게 한다. 카드의 기본적이 규격과 통신 형태는 ISO 7816 표준을 기본으로 정의된다[1,4,5].

2.1.2 스마트 카드의 하드웨어적 구조

스마트 카드는 자체연산 기능을 가능하게 하는 8bit-32bit의 CPU를 내장하고, CPU는 모토로라 6805이나 인텔 8051이 사용된다. 암호화 알고리즘을 통한 연산을 위하여 coprocessor라 불리는 암호 처리 전용 프로세서를 내장하고 있으며, 16KB-32KB 크기의 ROM과 EEPROM, RAM의 메모리를 지닌다. 사용자의 개인정보와 생성된 어플리케이션은 EEPROM에 저장되며, 칩이 제조되는 과정에서 COS가 ROM에 로드된다. COS는 EEPROM에 대한 접근을 제어하며 데이터를 보호하는 기능을 지닌다. RAM은 CPU의 작업공간으로의 역할을 한다.

2.1.3 스마트 카드의 어플리케이션 개발 환경

기존의 스마트 카드는 COS와 어플리케이션이 통합된 형태로 칩 의존도가 높았기 때문에 독자적인 어플리케이션의 개발이 어려웠다. 그러나 최근 COS와 어플리케이션의 분리를 통해서 다양한 어플리케이션을 개발할 수 있으며 Open Platform 형식의 멀티 어플리케이션이 가능하게 되었다. Open Platform Card로는 Java Card, Multos, WFS등이 개발되어 있다.

2.2 Java Card 기술

2.2.1 Java Card의 정의와 특성

Java Card는 기존의 스마트 카드의 장애요소를 보안하기 위한 기술로 Open Platform 형식의 개발 환경이다. Java Card는 1996년 Schlumberger의 산업센터에서 소개되었으며, 현재 Java Card 2.1환경을 지원하고 있다. Java Card는 다중 응용 어플리케이션을 위한 환경을 규격화한 것이며 독자적으로 어플리케이션의 개발이 가능하다. 개발자에게 사용이 용

이한 자바언어를 이용하여 프로그램을 구현하기 때문에 자바언어의 특징을 최대한 활용할 수 있으며, 자체적으로 제공하는 암호화 기능으로 안정성이 높다. 또한 Java Card에 저장되는 애플릿은 Java Card 플랫폼의 상부에 위치하게 때문에 한번 load된 애플릿은 재 컴파일의 요구가 없으므로 하드웨어적으로 독립성을 지니며, 여러 개의 어플리케이션을 관리하고 저장하는 다중 어플리케이션 기능을 제공한다. Java Card는 ISO 7816을 기본적인 기술로 적용하므로 스마트 카드의 다른 표준으로 정해져 있는 기술들과의 호환성도 뛰어나다[2,3,6].

2.2.2 Java Card의 구조

Java Card는 JCVM(Java Card Virtual Machine), JCRE(Java Card Runtime Environment), APIs (Application Programming Interfaces)로 구성되어 있다. JCVM은 생성된 소스를 구동하기 위한 것으로, 그림 1과 같이 off-card VM과 on-card VM으로 구성된다. off-card VM은 converter를 통하여 초기의 생성된 class file을 CAP file로 변형시켜 준다. CAP file은 class file을 압축한 형태이며 카드의 작은 메모리에 애플릿을 로드하기 위한 최소한의 데이터를 지닌다. CAP file은 인터프리터를 통하여 카드에 로드되어 인스톨된다.

JCRE는 Java card vendor로부터 애플릿을 분리하는 역할을 한다. 즉 하드웨어와 애플릿의 분리를 통하여 어플리케이션의 개발을 용이하게 한다. Java Card는 API를 통해서 어플리케이션을 위한 패키지와 클래스를 정의한다. 그림 2는 JCRE에서의 구조도 이다. Java Card 인터프리터와 시스템 클래스를 가지고 있으며, 애플릿과의 분리된 모습을 보여준다[2,3,6].

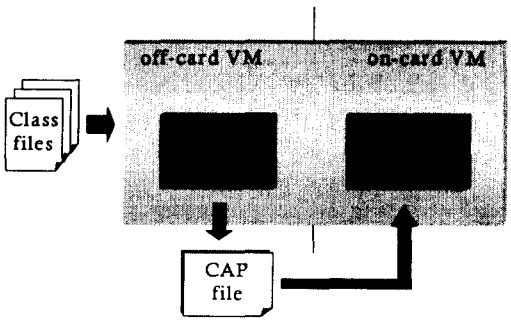


그림 1. Java Card Virtual Machine

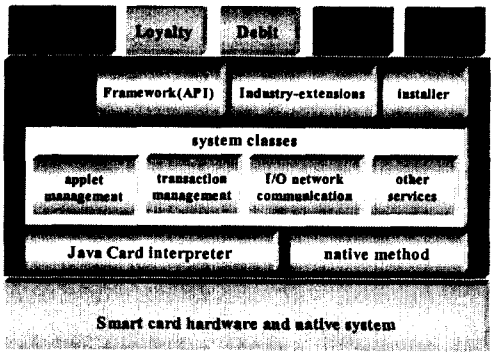


그림 2. Java Card Runtime Environment

3. 마일리지 관리 시스템

E-business상에서의 상업적 활동이 증대됨에 따라 사용자 위주의 서비스 제공 역시 활발히 이루어지고 있다. 특히 구매 과정에서 개별 쇼핑몰의 마일리지 지급을 통하여 사용자에게 보다 낳은 서비스를 제공한다. 거의 대부분의 쇼핑몰에서 마일리지를 지급하고 있으나 적립된 마일리지는 해당 쇼핑몰에서만 사용이 가능하다. 따라서 사용자들은 어디서나 마일리지를 사용할 수 있는 체계가 필요하게 되었다.

본 논문에서는 한국에서 성행하고 있는 마일리지 서비스와의 비교를 통하여 설명하도록 하겠다. 한국에서 활성화된 서비스로는 OKcashbag이나 easy-cash를 들 수 있다. 인기를 누리고 있는 서비스이기는 하지만 서비스들은 서버를 통한 관리이기 때문에 사용자에게 편리성을 제공하지 못한다. 따라서 본 논문에서는 기존의 시스템과 제안한 시스템을 비교 분석하여 보다 나은 시스템의 설계를 목적으로 한다. 본 시스템은 E-commerce상에서뿐만 아니라, M-commerce 상에서의 활용도 가능하다. 차세대 통신 서비스인 IMT-2000용 단말기에 USIM 카드를 기본으로 탑재하기 때문이다. USIM 카드는 스마트 카드의 새로운 이름으로 등장한 Java Card를 기반으로 어플리케이션을 개발하여 내장할 수 있다[7,8].

3.1 기존시스템과의 비교

그림 3은 기존의 시스템 구조도이며, 그림 4는 본 논문에서 제안한 시스템의 구조도이다. 기존의 시스템과 제안한 Java Card의 마일리지 통합 관리 시스템의 차이점은 서버의 유무에서 발견할 수 있다. 기

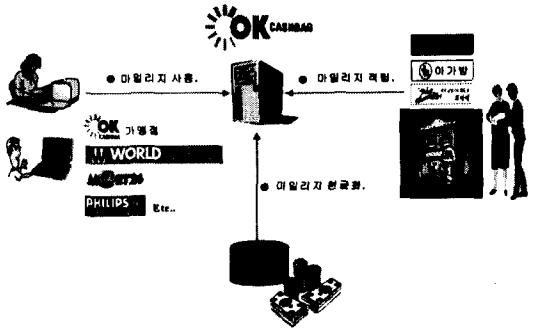


그림 3. 기존 시스템의 마일리지 시스템

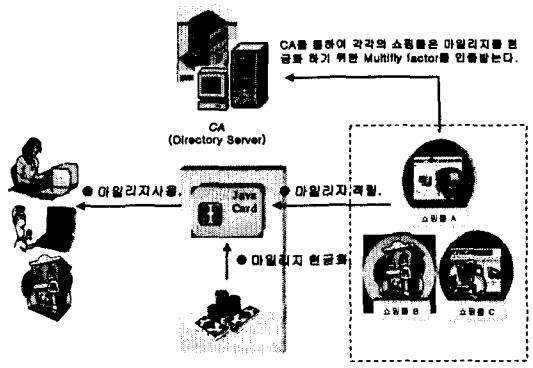


그림 4. Java Card를 이용한 마일리지 시스템

존의 시스템에서 제공되는 카드는 단지 카드의 번호만을 지정하는 빈 플라스틱 카드이다. 즉 물건을 구매한 후 지불 시에 카드를 제시하면 단말기를 통해 카드의 번호를 전송하여 시스템의 서버에 접속하게 된다. 즉, 그림 3에서와 같이 모든 서비스는 서버를 통해서 이루어지며 사용자의 데이터는 서버의 디렉토리에 저장되어 관리된다. 사용자가 데이터를 확인하기 위해서는 서버의 접속이 요구되며 서버의 접속을 통하여 사용자가 원하는 데이터를 확인할 수 있다. 보통 마일리지의 적립은 쇼핑몰이나 가맹점에서 이루어진다. 각 가맹점은 제공회사의 서버에서 인증을 하기 때문에 마일리지를 신뢰할 수 있으며 마일리지의 저장이 요구되는 경우 새로운 인증 과정은 필요하지 않으나 마일리지의 적립과 마일리지의 사용은 제공회사의 가맹점을 통해서만 가능하다. 적립된 마일리지는 일정금액 이상이 되어야 환급가능하며, 경우에 따라 서버에 직접 환급 신청을 해야만 환급이 가능하다. 시스템의 모든 단계는 항상 서버를 통해 관리되므로 일정기간과 노력이 소요된다.

본 논문에서 제안한 시스템은 Java Card에 저장되는 독립적인 프로그램으로 서버를 통한 관리에서 벗어나 사용자 관점에서의 관리를 의미하며, 본 시스템에서 필요한 모든 데이터는 Java Card의 메모리에 저장된다. 그림 4와 같이, CPU와 메모리로 구성된 스마트 카드를 사용함으로 Java Card내에 마일리지 관리 프로그램을 탑재한 후 자체적인 연산과정과 저장을 통하여 데이터를 관리할 수 있다. 마일리지의 적립은 인증기관에 등록된 모든 쇼핑몰에서 가능하며 인증기관의 인증을 통하여 안전성을 보장한다. 즉 마일리지를 전송 받을 경우 해당 쇼핑몰의 인증서를 포함하여 전송을 받거나, 만약 인증서의 용량이 클 경우에는 전자서명을 통하여 마일리지를 전송 받는다. 모든 인증은 인증기관을 통해 이루어지므로 신뢰할 수 있으며, 적립된 마일리지는 Java Card의 마일리지 관리 프로그램을 통해 즉시 현금화하여 사용할 수 있다. 기존의 빈 플라스틱카드의 번호만을 지니는 카드개념에서 벗어나 자체적으로 데이터를 저장하고 관리하게 때문에 시간의 소요 없이 바로 마일리지를 사용할 수 있으며 카드에 저장되어 있는 전자지갑과의 연계를 통하여 하나의 카드로 마일리지 적립과 현금의 사용이 가능하게 된다. 본 시스템의 또 하나의 큰 특징은 스마트카드 단말기만 있으면, 오프라인 상에서의 거래를 통한 마일리지 적립과 사용이 가능하다. 서버의 접속을 요구하지 않기 때문에 오프라인 상에서 쉽게 마일리지를 적립할 수 있다. 마일리지와 관련된 데이터를 스마트 카드에서 자체적으로 해결할 수 있는 특성 때문에, 서버를 통한 관리 시스템과 비교하여 사용자에게 편리성을 제공하며, 하나의 스마트 카드를 통해 여러 개의 어플리케이션을 사용할 수 있다는 장점을 지닌다. 표 1은 두 시스템의 차이점을 비교 분석한 결과를 보여준다.

3.2 제안한 시스템의 흐름도

그림 5는 본 논문에서 제안한 시스템의 흐름도 이다. 사용자와 쇼핑몰, 인증기관 사이의 관계와 마일리지를 적립하는 과정을 보여준다.

3.2.1 사용자

스마트 카드와 단말기를 통해 사용자는 쇼핑몰에 접속한다. 개인의 인증정보를 확인하기 위하여 인증

표 3. 기존의 시스템과 제안한 시스템의 비교분석

	기존의 시스템	본 연구에서 제안한 시스템
서비스 방법	서버를 통한 서비스	Java Card내의 마일리지 관리 프로그램
데이터 저장	서버의 디렉토리에 저장	Java Card의 메모리에 저장
제공하는 카드	빈 플라스틱의 카드로 번호만 지정	CPU와 메모리로 구성된 스마트 카드
데이터 관리	서버를 통한 데이터 확인, 관리	Java Card내의 프로그램을 통한 데이터 확인, 관리
마일리지 적립	카드 번호를 통한 서버 접속	직접 카드로 전송
현금화	누적포인트의 일정금액이상이 되어야 환급 가능 환급 신청을 해야만 환급 가능 항상 서버를 통한 과정으로 일정 기간이 소요	적립된 마일리지는 Java Card의 프로그램을 통하여 즉시 환급가능
마일리지사용	가맹점에서 현금과 마일리지를 동일하게 적용하여 사용 가능 신용카드를 사용할 경우, 신용카드와 본 시스템의 카드를 동시에 제시	Java Card내의 전자지갑을 이용하여 마일리지 사용 가능 전자지갑과 마일리지 프로그램이 동일한 카드 내에 저장
편리성	서버를 통한 관리로 일정한 시간과 노력이 필요	개인의 독립된 프로그램으로 사용자에게 편리성 제공
안전성	제공회사의 서버에서 각 가맹점의 마일리지 신뢰성 제공	인증기관을 통한 각 쇼핑몰의 factor 인증으로 신뢰성 제공
인증시점	제공회사의 인증으로 인증시점이 따로 필요하지 않음	마일리지와 각 쇼핑몰의 인증서를 함께 전송받음 인증서의 용량이 클 경우 각 쇼핑몰의 전자서명을 통하여 마일리지를 전송받음
안전성 보증	제공회사	인증기관
마일리지 적립과 사용 장소	제공회사의 가맹점	인증기관에 등록된 모든 쇼핑몰

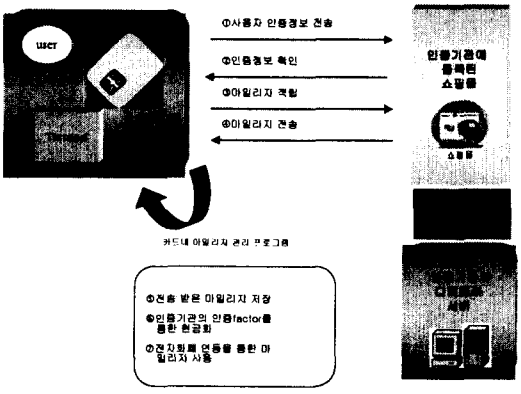


그림 5. 마일리지 관리 시스템의 흐름도

서를 전송한다. 인증정보의 확인이 완료되면, 사용자는 웹사이트를 통해 마일리지를 적립하고 카드의 마일리지 관리 프로그램으로 적립된 마일리지를 전송한다. 각 쇼핑몰에서 적립한 마일리지는 각각의 쇼핑몰마다의 값과 통합적으로 계산한 값을 모두 가지고 있어야한다. 각 쇼핑몰에서의 마일리지 적립금에 따라 회원 자격이 다르게 주어지기 때문이다.

3.2.2 쇼핑몰

각 쇼핑몰에서 사용하는 마일리지 factor는 인증기관을 통해 인증을 받는다. 따라서 마일리지의 안정성은 인증기관을 통해 보장받을 수 있다.

3.2.3 인증기관

인증기관은 각 쇼핑몰의 마일리지 factor를 인증하는 곳으로 디렉토리 서버를 통해 각 쇼핑몰의 인증 데이터를 관리한다.

4. 시뮬레이션

본 시스템의 유효성을 증명하기 위하여 프로그램을 구현하였다. J빌더 상에서 자바언어를 사용하여 프로그래밍 하였으며, 카드와 단말기 사이의 통신을 위하여 APDU를 사용하였다. 카드와 어플리케이션은 command와 response 명령을 이용하여 통신을 하

표 2. 본 시스템의 AID와 주요 Method

AID	A0 00 00 00 18 FF 00 00 00 00 00 00 00 00 01 02	
	Method	Command
Send Mileage	send()	00 31 00 00 02 00 05 02
Sum of Mileage	getBalance()	00 30 00 00 02 10 00 02
Save Mileage	save()	00 32 00 00 02 00 10 00
Verity PIN	verifyPin()	00 33 00 00 04 01 02 03 04 00

며, 생성된 애플릿은 고유의 AID(Application Identifier)를 갖게 된다. 카드와 단말기 사이의 접속 시에는 항상 인증 단계가 요구된다. 각각의 쇼핑몰은 임의로 정하여 애플릿으로 구현하였으며 TCP/IP 통신 프로토콜을 이용하여 카드내의 마일리지 관리 시스템과 통신을 한다. 각 쇼핑몰의 마일리지 factor는 카드의 메모리에 저장되어 있으며, 그 값을 통해 각 쇼핑몰의 마일리지를 현금화한다. 적립된 마일리지는 사용자가 항상 볼 수 있어야 하며 Wallet 과 같은 전자화폐 시스템과의 연동을 통하여 마일리지를 바로 현금으로 사용할 수 있도록 한다[9].

표 2에서 보는바와 같이 Java Card에 저장되는 프로그램은 각각의 AID를 지닌다. AID를 통해 애플릿의 유일성을 입증한다. 본 시스템의 AID는 'A0 00 00 00 18 FF 00 00 00 00 00 00 00 00 01 02' 로 주었으며, 애플릿에 주요 메소드의 command를 주었다. Java Card의 애플릿은 APDU의 command와 response 명령을 주고받으며 통신을 한다.

표 3은 마일리지 프로그램이 Java Card에 인스톨되는 과정을 보여준다. Java Card상에서 마일리지 프로그램을 사용하기 위해서는, Java Card의 메모리에 프로그램을 인스톨해야 한다. 인스톨 메소드는

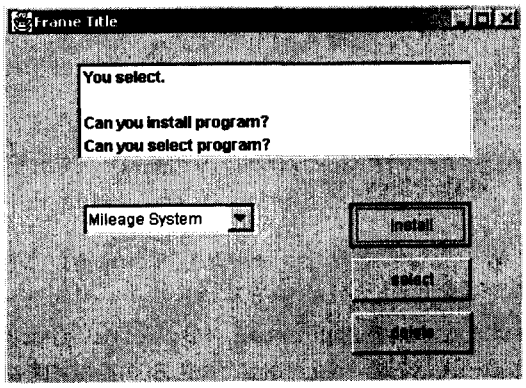


그림 6. 마일리지 프로그램의 인스톨

표 3. 인스톨된 프로그램의 크기

```
Install start.....                                  Length :
Starting load process...                            1794Jar to Hex> OK
Load in the card...                                 block 1 loaded
Path : mileage/javacard                             block 2 loaded
mileage/javacard/Header.cap size = 29....ok         block 3 loaded
mileage/javacard/Directory.cap      , size = 34....ok   block 4 loaded
mileage/javacard/Import.cap size = 24....ok         block 5 loaded
mileage/javacard/Applet.cap size = 23....ok         block 6 loaded
mileage/javacard/Class.cap , size = 23....ok        block 7 loaded
mileage/javacard/Method.capsize = 1060....ok        block 8 loaded
mileage/javacard/StaticField.cap    , size = 13....ok   block 9 loaded
mileage/javacard/ConstantPool.cap   , size = 149....ok  package loaded in 5 s
mileage/javacard/RefLocation.cap    , size = 130....ok  Loading mileage succeed
mileage/javacard/Descriptor.cap     , size = 309. ok    Install applet...
                                                    Installing application mileage succee
```

Java Card의 framework에서 제공되는 메소드이며, 이 메소드를 사용하여 프로그램을 인스톨 할 수 있다. 그림 7은 프로그램이 인스톨 될 때 CAP 파일의 크기를 보여주는 그림이다. Java Card의 특성상 메모리가 크지 않기 때문에 Java Card에 저장되는 파일은 class 파일을 압축한 형태의 CAP 파일이다. Java Card의 메모리에 저장되는 CAP 파일의 최종 크기는 1794kbyte이다.

Java Card 상에서 프로그램을 실행하기 위해서는 선택 메소드를 통해 프로그램을 선택해야 한다. 그림 7은 프로그램을 선택하는 과정을 시뮬레이션 하는 그림이다. Java Card의 메모리에는 여러 개의 프로그램이 인스톨되어 있으므로 선택 메소드를 통해 사용할 프로그램을 선택해야 한다.

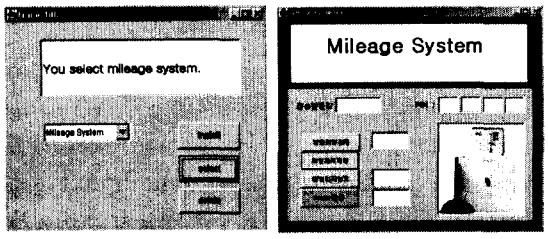


그림 7. 마일리지 프로그램의 사용을 위한 select

표 4. 인증과정과 초기화 과정

```
ATR returned by the Card...
<-- Card : 3B 6E 00 00 80 31 80 65 B0 03 02 01 5E 83 00 00 90 00
cardRandom: 86 DB BF 9D A6 67 28 18
hostRandom: 00 00 00 00 00 00 00 00
derivationInputData: A6 67 28 18 00 00 00 00 86 DB BF 9D 00 00 00 00
Encryption staticKey: CA CA CA CA CA CA CA CA 2D 2D 2D 2D 2D 2D 2D CA CA CA CA CA CA CA CA
Encryption sessionKey: A4 7E 16 D7 43 49 69 EA FA EC 5F 2D 86 CF B7 67 A4 7E 16 D7 43 49 69 EA
Macing staticKey: 2D 2D 2D 2D 2D 2D 2D 2D CA CA CA CA CA CA CA CA 2D 2D 2D 2D 2D 2D 2D 2D
Macing sessionKey: 87 7D D3 6C 3D A8 0F 70 0D 75 F3 47 E3 18 E7 9F 87 7D D3 6C 3D A8 0F 70
KEK staticKey: CA 2D CA 2D CA 2D CA 2D 2D CA 2D CA 2D CA 2D CA CA 2D CA 2D CA 2D CA 2D
KEK sessionKey: CA 2D CA 2D CA 2D CA 2D 2D CA 2D CA 2D CA 2D CA CA 2D CA 2D CA 2D CA 2D
hostAndCardRandom: 00 00 00 00 00 00 00 00 86 DB BF 9D A6 67 28 18 80 00 00 00 00 00 00 00
calculatedCardCryptogram: B8 C7 3C B9 97 4A BD D4
cryptoFromCard: B8 C7 3C B9 97 4A BD D4
cardAndHostRandom: 86 DB BF 9D A6 67 28 18 00 00 00 00 00 00 00 00 80 00 00 00 00 00 00 00
hostCryptogram: 9E 6B 01 44 5C 23 43 11
Authentication OK
Initialize OP global PIN
PIN initialisation OK
Select application...
Select application null OK
```

표 4는 카드에 저장되어 있는 프로그램 중 마일리지 프로그램을 선택하는 과정이다. 주어진 AID를 비교하여 인증과정과 초기화 과정을 거친 후 어플리케이션에 접근한다.

그림 8은 마일리지를 적립하는 과정을 보여준다. 각 쇼핑몰의 마일리지 factor는 Java Card의 메모리에 저장되어 있다. 전송 받은 값은 각 쇼핑몰마다의 factor를 이용하여 마일리지로 변환한다. 각각의 쇼핑몰을 마일리지를 통하여 회원자격을 부여하기 때문에 쇼핑몰마다의 마일리지 값은 사용이 되었어도 적립한 상태의 값을 보존하고 있어야 하며, 각 쇼핑몰마다의 마일리지의 총합 또한 계산이 되어야 한다. 그림 10에서 보여지는 코드는 터미널과 카드 사이의 APDU 명령어을 나타낸다. command 명령과 response 명령을 주고받으며 response 명령어가 '90 00'

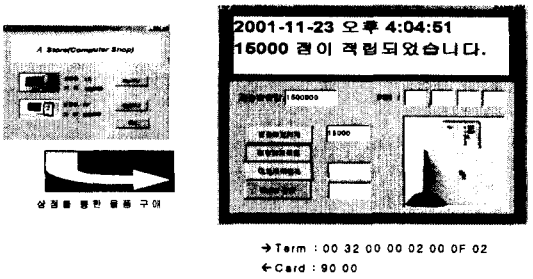


그림 8. 마일리지 적립 단계 시뮬레이션

이란 값을 지니면, 그 과정이 성공했다는 것을 의미한다.

그림 9은 현재 적립되어 있는 마일리지를 확인하는 과정과 Wallet으로 전송하는 과정을 보여주는 그림이다. 적립된 마일리지는 wallet과 같은 다른 어플리케이션으로 전송하여 사용한다. wallet으로 전송하는 것은 화폐의 가치를 인정하는 것이기 때문에 PIN을 부여함으로 보안성을 높인다. 마일리지 합계 버튼을 이용하여 카드 내에 저장되어 있는 마일리지의 합계를 확인할 수 있다.

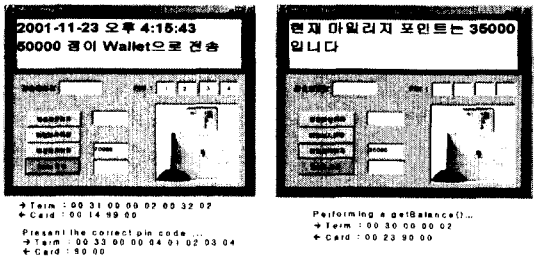


그림 9. 마일리지 확인과 Wallet 전송 단계

5. 결 론

자바카드 상에서의 효율성 있는 마일리지 관리 시스템을 제안하였다. 스마트 카드의 차세대 COS로 주

목받는 Java Card는 어플리케이션 개발의 용이성과, 뛰어난 독립성을 바탕으로 효율적인 시스템을 개발할 수 있다. 본 논문에서는 기존의 마일리지 서비스와의 차이점을 비교 분석하여 유용성이 높은 새로운 시스템을 제안하였다. Java Card 내에 저장되는 한 개인의 독립적인 프로그램으로, 카드의 메모리 상에 개인의 인증정보와 쇼핑몰의 마일지리 factor등의 데이터를 저장하고, CPU를 통한 연산작용을 이용하여 마일리지를 적립하고 현금화하여 사용할 수 있다. 즉 기존의 서버를 이용한 서비스와 비교하여 처리속도나 시스템 부하, 구축비용 등을 절감하는 효과를 가져올 수 있다. 본 시스템은, 스마트 카드에서의 활용뿐 아니라, IMT-2000 서비스 상에서의 단말기에 탑재되는 USIM 카드에도 적용이 가능하다.

스마트 카드의 시장은 시작단계로 활성화되고 있는 추세이기 때문에, 다양한 어플리케이션의 개발이 미흡한 실정이다. 따라서, 다양한 어플리케이션의 개발이 뒷받침되어야 할 것이며, 생체인식 및 bluetooth 기술과의 접목을 통하여 카드의 활용성을 높이도록 해야 할 것이다.

참 고 문 헌

[1] Wolfgang Effing and Wolfgang Rankl, "Smart Card Handbook", John Wiley & Sons, 2000.

[2] Zhiqun Chen, "Java Card Technology for Smart Cards", Addison-Wesley, 2000.

[3] Eric Vetillard, "Java Card 2.1 general presentation", Gemplus Developer Conference, 1999.

[4] C. Schnorr, "Efficient Signature Generation by Smart Cards", Journal of Cryptology, Vol. 4, No 3, pp. 161-174, 1991.

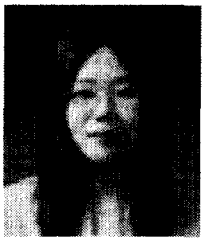
[5] P. Biget, P. George, and J. Vandewalle, "How Smart-Cards Can Take Benefits From Object-Oriented Techonogies", In Hartel et al, pp. 175-194, 1999.

[6] 하남수, 홍인식 " IMT-2000에서의 USIM을 위한 구조 설계 및 응용 프로그램 구축에 관한 연구", 정보처리학회 춘계학술대회, 제 8권 제1호, pp. 627-630, 2001.

[7] 백장미, 강병모, 홍인식, "Java Card를 이용한 인터넷 쇼핑몰 마일리지 통합 관리 시스템에 관한 연구", 정보과학회, 제28권 제2호, II, pp. 214-216, 2001.

[8] 강병모, 백장미, 홍인식, "IMT-2000 단말기 상에서의 USIM카드를 이용한 쇼핑몰 마일리지 통합 관리 시스템에 관한 연구", 정보처리학회, 제8권 제2호, pp. 1423-1426, 2001.

[9] Ivor Horton, "Beginning Java2", WROX, 2000.

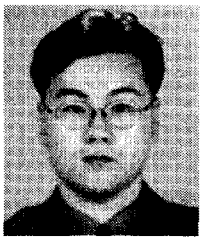


백 장 미

2001년 순천향대학교 컴퓨터학부 (학사)

2002년 순천향대학교 대학원 전산학과 석사과정

관심분야 : M-commerce, 스마트 카드, 모바일 통신



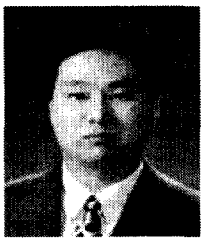
강 병 모

1998년 순천향대학교 전산학과 (학사)

2001년 순천향대학교 대학원 전산학과 (석사)

2002년 순천향대학교 대학원 전산학과 박사과정

관심분야 : 모바일 통신, 멀티미디어



홍 인 식

1981년 한양대학교 전자공학과 (학사)

1986년 한양대학교 대학원 전자공학과(석사)

1988년 한양대학교 대학원 전자공학과(박사)

1991년~1995년 순천향대학교 공과대학 전산학과 전임강사

1995년~1999년 순천향대학교 공과대학 컴퓨터학부 조교수

1999년~순천향대학교 공과대학 정보기술학부 부교수

관심분야 : 임베디드 시스템, 스마트 카드, 모바일 통신