J2ME 기반 유·무선 연동의 모바일 전자지갑 설계 및 구현

박 남 제† · 송 유 진††

요 약

무선인터넷의 활성화와 더불어 기존의 유선 및 무선 전자상거래를 통합하는 여러 방안에 대한 관심이 고조되고 있다. 유·무선 통합형 전자상거래를 위해서 무선통신 환경에서의 최적화된 보안 및 인증 그리고 사용자에게 보다 편리한 사용방안이 선결되어야 한다. 본 논문에서는 J2ME(Java 2 Micro Edition) 기반 MIDP(Mobile Information Device Profile)를 이용해서 전자지불용 전자지갑 모듈 개발을 통해 유선과 무선에서 공통으로 사용할 수 있는 유·무선 연동의 모바일 전자지갑(Mobile Wallet)을 설계 및 구현하였다. 본 논문에서 구현한 전자지갑의 특징은 지불정보의 암·복호화를 통한 안전성 제공과 전자지갑의 다운로드를 통해 유·무선 환경에서 온라인 형태로 사용할 수 있는 기능을 제공하는 것이다. 또한 유·무선인터넷 전자상거래에서 안전하고 편리한 지불방안을 제시한다.

Design and Implementation of J2ME-based Wired/Wireless Interworking Mobile Wallet

Nam Je Park† · You Jin Song††

ABSTRACT

Together with the activation of wireless Internet, the interest for various integrated methods of the existing wired and wireless e-commerces is on the rise. For the integrated wired and wireless e-commerce, firstly, the optimized security and authentication under the radio communication environment should be decided. And also the user-friendly usage is important. In this paper, we designed and implemented wired and wireless interworking Mobile Wallet on terminal that can be used through both wired and wireless methods by developing electronic wallet module for electronic payment using MIDP (Mobile Information Device Profile) based on J2ME (Java 2 Micro Edition). The characteristics of mobile wallet implemented in this paper provide the stability through encoding/decoding payment information and on-line usage on wired/wireless environments through downloading electronic wallet. In addition, we also present safe and convenient payment method for e-commerce based on wired/wireless Internet.

키워드 : 모바일 전자지갑(Mobile Wallet), M-Commerce(Mobile Commerce), 전자지불(Electronic Payment), J2ME(Java 2 Micro Edition)

1. 서 론

무선 인터넷 서비스는 언제 어디서든지 원하는 서비스를 받을 수 있는 환경을 제공함으로써 사용자로 하여금 지불에 대한 거부감을 줄이고, 다양한 컨텐츠를 사용자에게 제공함으로써 서비스 및 망 사업자에게 고객 확보와 수익성 증대의 효과를 가져온다. 무선 인터넷 서비스의 확산으로 가장 이슈화되는 분야 중 하나가 전자지불이다. 서비스 사업자들은 새로운 서비스를 제공하는 제품과 다양한 지불 방법을 통한 높은 수익을 올리기 위해 보다 유연하고 확장성 있는 전자지불 체계를 요구하고 있다.

무선 인터넷을 기반으로 하는 무선 전자상거래는 기존 유선 전자상거래에서 제공하기 힘들었던 이동성(mobility), 편재성(ubiquity), 그리고 이로부터 발생하는 위치 기반서비스(Location Based Service) 제공이 가능한 여러가지 이점이 있다. 이와 함께 무선인터넷 서비스의 요구사항으로써 상호운영성(interoperability), 확장성(scalability), 효율성(efficiency), 신뢰성(reliability) 및 보안성(security)을 고려하여야 한다[1]. 특히 무선 인터넷에서의 정보보호는 전송계층 및 응용계층에서 접근이 이루어져야하고 무선 환경의 제약사항을 고려하여야 한다. WAP(Wireless Application Protocol) 방식인 경우, 무선 게이트웨이로 인해 종단간 보안을 제공하기가 어렵다는 문제가 있으며 이를 해결할 수 있도록 해야 한다. 또한

※ 본 연구는 동국대학교 논문게재연구비 지원으로 이루어졌음.
† 정 회 원 : 성균관대학교 정보통신대학원 정보보호학과
†† 정 회 원 : 동국대학교 정보산업학과 교수
논문접수 : 2002년 7월 29일, 심사완료 : 2002년 10월 24일

무선 전자상거래 환경에서 다양한 응용서비스를 제공하기 위해 플랫폼 독립적인 어플리케이션 운영 기능을 제공해야 한다.

무선 플랫폼상의 WAP 게이트웨이에서의 보안의 취약점이라든지, 종단간 보안 문제를 해결하는 방안중의 하나인 J2ME 환경에서 지원 플랫폼 자체에서 지원하는 보안기능을 이용하여 응용 계층에서의 WPKI(Wireless Public Key Infrastructure) 구조를 새로이 만들어 사용할 수 있다. 본 논문에서는 J2ME 상에서 동작하는 지불 보안 프로토콜을 적용한 보안 방식을 사용하여 모바일 전자지갑을 설계 및 구현하였다.

보안상 매우 취약한 신용카드를 대체할 수 있는 휴대폰 기반 오프라인 지급 결제를 수행하는 전자지갑 모델은 기존 신용카드의 문제점을 극복하는 편리한 방법을 제공한다. 이러한 관점에서 본 논문에서는 모바일 플랫폼 기반의 전자지불 기능을 구현할 수 있는 방안을 제시한다. 즉, 무선 인터넷 응용 프로토콜인 WAP, ME(Mobile Explorer) 등을 중심으로 모바일 플랫폼과 연결한 단말기 브라우저를 탑재하여 유·무선 전자상거래에서의 전자지불을 위한 모바일 전자지갑의 구현 방안을 연구하였다. 모바일 전자지갑을 구현함에 있어 전자지갑 모듈의 설계와 동작 프로세스에 대해 설명하며, 모바일 전자지갑을 이용한 유·무선 전자상거래 서비스에 대하여 휴대폰 단말기의 사용자 인터페이스 구현 화면을 설명한다.

본 논문의 구성은 다음과 같다. 2장에서 관련연구, 3장에서 모바일 전자지갑의 요구사항을 분석하고, 이를 바탕으로 4장에서 지불 보안프로토콜 및 모바일 전자지갑 설계에 대하여 기술한다. 5, 6장에서 실제 구현한 플랫폼상의 전자지갑 화면을 설명하고, 안전성 분석과 성능평가를 기술한다. 마지막으로 7장에서 결론을 맺는다.

2. 관련 연구

2.1 J2ME

자바 모바일 플랫폼은 이동통신 단말기들의 작은 디바이스들을 위한 플랫폼으로 동적 어플리케이션 다운로드 서비스 기능 제공, 플랫폼 간의 호환성 제공, 향상된 인터페이스 표현, 네트워크 환경과의 비연결성 고려, 무선 환경의 종단간 보안 해결책 등의 장점이 있다. 구성요소인 CLDC(Connected Limited Device Configuration)의 특징은 부동 소수점을 지원하는 것은 오버헤드를 발생하기 때문에 지원하지 않고, 가비지 컬렉션을 지원하지 않는다[9]. 예외 처리를 지원하지만 임베디드 환경에 의해 제한된 에러 처리만 하고, 네이티브 함수를 호출하는 JNI는 지원하지 않는다. MIDP(Mobile Information Device Profile)는 Java API의 한 묶음으로, CLDC와 함께 셀룰러 폰, 양방향 삐삐와 같은 이동 정보 단말기를 위한 J2ME 어플리케이션 운영 환경을 제공한다. JAM(Java Application Manager)은 CLDC/MIDP 플랫폼의 새로운 어플리케이션 모델을 지원하기 위한 어플리케이션 관리 소프트웨어이고, 그 역할은 MIDP 어플리케이션인 미들릿(MIDlet)을 다운로드하여 설치, 업그레이드, 실행, 삭제하는 것이다.

2.2 PACS 프로토콜

PACS(Personal Communications Services) 프로토콜은 선택적으로 공개키 기반 인증 프로토콜을 지원하는 키 분배 프로토콜이다. 이 방식은 기지국에서 암호화 및 서명없이 메시지가 전송되므로 인증 및 비밀성, 부인봉쇄가 불가능하게 된다. 동시에 기지국에서 암호화된 정보를 받는다 할지라도 별도의 임의의 값이 사용되지 않으므로 키 일회성은 획득할 수 없는 단점을 가지고 있다[13].

2.3 ASPeCT 프로토콜

ASPeCT 프로토콜은 EC(European Commision) ACTS project ASPeCT에 의해서 개발되었고, UMTS(Universal Mobile Telecommunication System)에서 사용자와 서비스 제공자인 VASP(Value Added Service Provider) 사이의 공개키 기반 인증 및 키 설정 프로토콜 중의 하나이다.

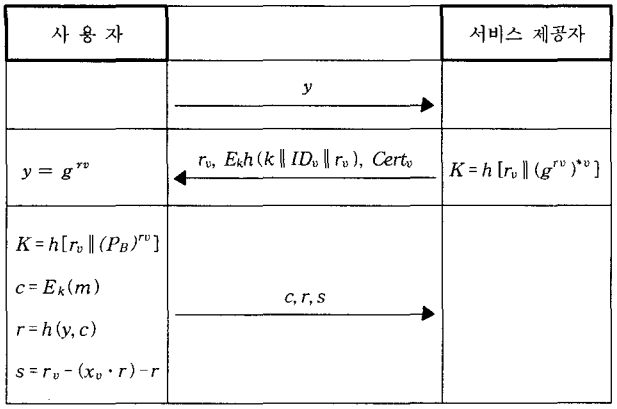
공개키 암호방식을 기본으로 하는 ASPeCT 프로토콜은 기존의 이동통신 시스템에서 제공하는 인증 프로토콜의 기밀성, 시스템의 부정 이용을 방지하는 네트워크에 의한 사용자 인증 외에 공격자의 네트워크 위장을 방지하기 위해 사용자에 의한 네트워크 인증과 프로토콜 내에 과금 관련 데이터를 통합하여 서비스 이용에 대한 이용자 검증 및 서비스 부인방지 등의 보안 특성을 더욱 강화하였다[2]. 이 프로토콜은 상호개체인증, 명확한 키 인증, 키 동의, 키의 신규성에 대한 확인을 서로에게 확인을 만족시키지만, 익명성과 부인봉쇄는 송신자에게만 만족시키는 단점이 있다[11].

2.4 경북대 AKA 프로토콜

경북대에서 제안한 AKA 프로토콜은 이동 통신이 갖는 제약점인 전력 소모량, 메모리 크리, 디스플레이 크기, 전송 속도, 안정성, 대역폭 등을 해결하기 위해 빠르고도 안전한 공개키 기반 알고리즘인 XTR-Signcryption(Gamage, Leiwo and Zheng's signcryption version)을 이용하였다[7, 11]. 제안한 XTR-Signcryption 기반의 인증 및 키 합의 프로토콜은 Modified Singcryption이 갖는 암호화와 서명이 한번에 이루어지는 특징과 XTR이 갖는 연산량 감소와 작은 키 길이로 인해 무선 환경에 적합하다[8].

이 프로토콜은 이동 단말기 초기화 단계에서 주요 변수들이 설정되며 서비스 요청시에 필요한 $y = g^{r_U}$는 실 시행 단계에서의 계산량을 줄이기 위해 사전 계산 단계에서 연산된다. 이동 통신 사용자가 서비스 제공자로부터 서비스를 제공받고자 할때, 사전 계산 값인 y를 서비스 제공자에게 전송함으로써 프로토콜이 진행된다. 프로토콜 실 시행 단계에서는 서비스 제공자의 세션키 계산, 서비스 제공자 정보 암호화, 암호화된 값을 이동 통신 사용자에게 전송, 이동 통신 사용자의 세션키 계산, 서비스 제공자로 받은 정보 복호, 서비스 제

공자의 정보 검증, 사용자 정보 암호화, 암호화된 정보 전송, 서비스 제공자에서의 정보 수신, 수신된 정보 복호, 복호된 사용자 정보 검증이 이루어진다[11](그림 1).



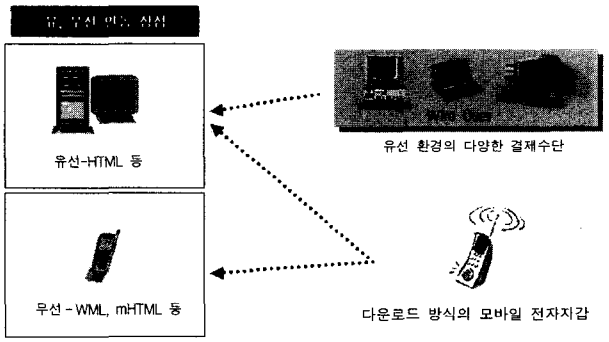
(그림 1) 경북대 AKA 프로토콜

3. 유·무선 연동 모바일 전자지갑의 요구사항

본 장에서는 모바일 전자지갑의 설계 및 구현을 위한 기본 요구사항 및 안전한 지불에 대한 요구사항에 대해 기술한다.

3.1 일반 요구사항

정보 통신기술의 발달에 따라 유무선 영역 구분이 사라지면서 신기술의 등장이 아닌 기존 기술통합을 통한 새로운 복합 서비스의 형태로 구성되는 유·무선 연동 솔루션은 전자상거래의 주요한 기능을 제공할 수 있다. 유선과 무선 각각의 플랫폼에서 운영되는 전자지갑의 한계점을 유·무선 연동의 모바일 전자지갑을 통해 극복할 수 있다. 우선 모바일 서비스 플랫폼상의 다운로드 어플리케이션 방식을 기반으로 하는 개방형 플랫폼상의 환경을 적용하여 상호 인증 프로토콜을 통해 유·무선 연동이 가능한 상점으로부터 구매한 상품을 모바일 전자지갑에서 지불할 수 있다(그림 2). 이러한 특성을 통해 유·무선 연동이 가능한 모바일 전자지갑의 일반 요구사항이 만족된다.



(그림 2) 유·무선 연동방식의 전자지갑 개념도

유·무선 연동 지불 보안 프로토콜 설계에서의 요구사항은 무선 환경에 적합한 통신 패스의 최소화, 대역폭 사용의 효율화, 연산 부하의 최소화의 조건이 전제되어야 한다. 따라서 이동통신이 갖는 제약점인 저용량의 메모리와 같은 제한된 자원, 제한된 계산력, 제한된 대역폭에 적합한 프로토콜이 설계되어야 한다.

3.2 보안 요구사항

무선인터넷 보안의 기본적인 사항인 기밀성(confidentiality), 인증(authentication), 무결성(integrity), 부인방지(non-repudiation), 가용성(availability), 접근통제, 적은 계산량 등의 사항들이 고려되어야 한다. 이러한 사항들을 기반으로 하여 지불 프로토콜의 보안 요구사항을 살펴보면 다음과 같다.

(1) 상호 개체 인증
서로 다른 개체에 대한 인증이 필요하다.

(2) 공개키 인증서의 상호 교환
사용된 공개키의 정당성에 대한 확인 과정을 위해 공개키 인증서를 상호 교환한다.

(3) 세션키에 대한 상호 동의
교환 데이터를 보호하기 위해 세션키는 상호 각각이 생성하여야 한다.

(4) 세션키에 대한 상호 제어
서로의 정보가 세션키에 미치는 영향이 동일하여야 한다.

(5) 키 인증
함축적(implicit) 키 인증은 통신에 참여하지 않는 다른 개체가 설정된 세션키를 얻을 수 없도록 하기 위해 대응되는 키를 가능한 참여하는 개체만이 계산할 수 있어야 하고, 명시적(explicit) 키 인증은 대응되는 키를 가능한 한 참여하는 개체만이 계산할 수 있어야 하고 실재로도 계산되어야 한다.

(6) 상호 키 확인
서로의 세션키가 실제 같은 키를 소유하고 있다는 확인이 있어야 한다.

(7) 키의 신규성에 대한 상호 보증
이전 메시지의 재사용으로 이전의 키를 재설정하는 것이 방지되어야 한다.

(8) 지불정보의 기밀성
데이터의 가로채기를 방지하기 위해 주고받는 데이터는 암호화되어야 한다.

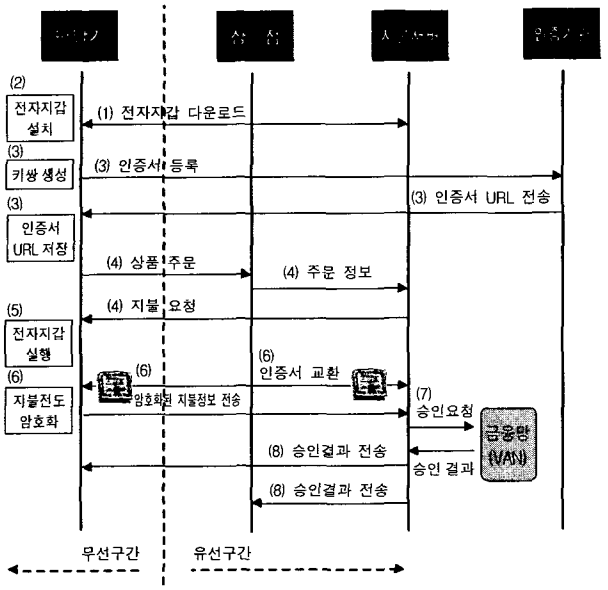
(9) 부인 봉쇄
중요한 데이터의 송·수신에 대한 부인 방지 기능이 보장되어야 한다.

4. 모바일 전자지갑의 설계

본 장에서는 유·무선 전자상거래에서의 모바일 전자지갑과 지불서버와의 프로토콜에 대한 설계 및 사용자 인터페이스, 기능 설계에 대해 기술한다.

4.1 전자지갑방식의 지불 흐름도

모바일 전자지갑 방식은 사용자가 직접 전자지갑에 지불수단 정보(신용카드 번호나 유효기간, 선불형 전자화폐 정보, 계좌정보)를 입력하여 전자지갑에 등록하고 상품을 구매한 후 등록되어 있는 정보를 선택하여 지불을 수행하게 된다. 전자지갑 기반의 인증 및 지불 흐름도는 (그림 3)과 같다.



(그림 3) 모바일 전자지갑 방식의 지불 흐름도

전자지갑을 이용한 무선인터넷 환경에서의 인증 및 지불 흐름도는 (그림 3)과 같으며, 그 구성은 다음과 같다.

(1) 사용자는 지불을 위해 모바일 전자지갑을 단말기로 다운로드 한다.
(2) 사용자는 모바일 전자지갑에 사용자 등록 과정을 수행한다.
(3) 키 쌍을 생성하여 인증기관으로부터 인증서를 발급 받는다(WPKI 연동).
(4) 사용자는 상점에 접속하여 구매과정을 마친다.
(5) 사용자는 지불을 위해 모바일 전자지갑을 구동한다.
(6) 사용자의 지불 정보를 암호화하여 지불 시스템으로 전달한다.
(7) 지불서버는 시스템은 지불정보를 각 금융기관에 전달하고, 승인 정보를 받는다.
(8) 지불서버는 승인 정보를 상점에게 전달한다.
(9) 상점은 구매자에게 승인정보 전달과 상품배달을 시행한다.

모바일 전자지갑은 자체 DB를 이용하여 무선 전자지갑 비밀번호, 신용카드 정보, 인증서 URL 등의 정보를 저장하고 있으므로 외부 망으로의 정보 유출의 차단 및 단말기 분실에 따른 개인 정보의 침해를 막을 수 있다.

4.2 인증 및 지불보안 프로토콜의 설계

본 장에서는 실제 구현할 유·무선 연동의 모바일 전자지갑의 지불 보안프로토콜 설계에 대해 살펴본다.

본 논문에서의 구현되어진 프로토콜은 모바일 전자지갑과 지불서버간에 무선 통신이 이뤄지므로 가능한 각 측면에서 연산량과 그로 인한 부하를 줄여야 한다. 기존에 사용된 인증 및 키 교환 프로토콜은 D-H(Diffie-Hellman) 프로토콜 기반하에 세션 키를 설정하므로 연산량이 많게 된다. 3세대 이동통신 시스템에서는 성능이 좋은 CPU와 메모리를 사용하므로 이것보다 연산에 걸리는 시간이 줄어들겠지만, 휴대용 단말기에서의 연산량이 많으므로 실시간 인증을 필요로 하는 이동 통신 시스템에서는 실현가능성이 어렵게 된다.

제한된 대역폭과 전송 속도, 제한된 계산 능력, 전력 소모량, 메모리 크기 등을 고려해야 하므로 보다 짧은 키 길이를 요구하게 된다. 여기에서 효율성을 위하여 키 길이를 줄이기 위해 적용시킬 새로운 암호 시스템에 관한 연구가 필요하다. 그 대표적인 예가 타원 곡선, NTRU, XTR 등이다[8].

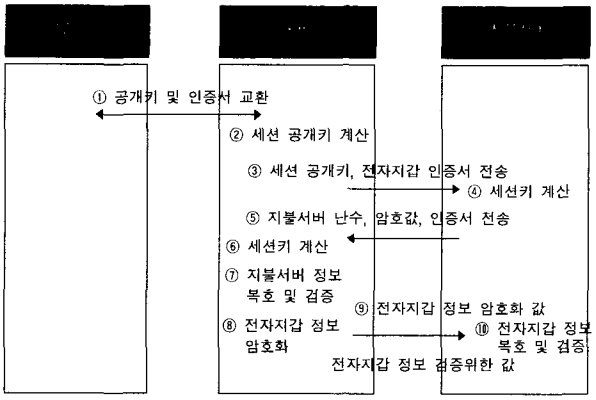
우선 타원 곡선에 대해 살펴보자. 타원 곡선은 군(Group)을 제공할 수 있는 다양한 타원곡선을 활용할 수 있다. 그리고 초특이(Supersingular) 타원곡선을 피하게 되면 이 군에서의 Subexponential Time Algorithm이 존재하지 않는다. 또한 타원 곡선 암호시스템은 존재하는 다른 공개키 스킴과 같은 안전도를 제공하는 데에 더 작은 키 길이를 가지고 가능하다. 이러한 타원곡선 암호 시스템의 특징 때문에 무선 인터넷 보안 솔루션으로 각광을 받고 있지만 키생성 시간이 긴 단점이 있다. 구현 방법에 따라 달라지지만, 일반적으로 키 생성 시간이 약 5초 정도 소요된다고 알려져 있다. 이런 문제 때문에 타원 곡선 암호 시스템보다 키 생성 시간을 줄일 수 있는 암호 시스템이 무선 환경에는 더 적합할 것이다.

NTRU는 격자 줄임(lattice reduction)에 기반을 둔 암호 시스템이다. 이 시스템의 암호 단계에서는 다항식 대수와 두 개의 모듈라 q와 p의 줄임(reduction)의 결합 시스템으로 사용한다. 그리고 복호 단계에서는 기초적인 확률이론에 의존하는 비결합 시스템으로 사용한다. NTRU 공개키 암호 시스템의 보안성은 모듈라 q와 p의 독립된 줄임을 갖는 다항식 혼합 시스템에서의 상호 작용으로부터 온다. 그리고 큰 격자(lattice)에서 지극히 짧은 벡터를 찾아낸 것이 어렵다는 문제에 의존한다. 이 NTRU는 암호와 복호가 빠르며, 쉽다. 길이 N의 메시지 블록(block)를 암·복호하는데 사용된 연산량은 RSA가 $O(N^3)$인데 반해, NTRU는 $O(N^2)$이다. 이런 특징 때문에 무선 환경에 적합하다[12].

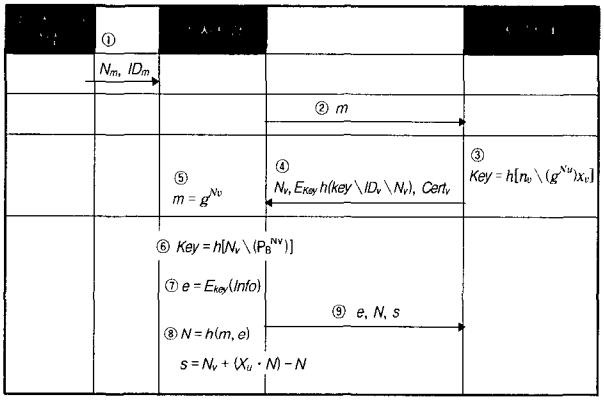
XTR은 새로운 암호시스템이 아니라 유한체의 승법군의 부분군을 사용한 전통적인 이산대수 시스템이다. XTR은 LUC처럼 부분군의 원소를 표현하기 위한 효율적이고 간결한 새로운 방법을 사용한다. XTR의 시큐리티(security)는 $p^2 - p + 1$을 나누는 위수의 $GF(p^6)^*$의 부분군에서 이산 대수 문제

에 기반한다(LUC는 $p+1$을 나누는 위수의 $GF(p^2)^*$의 부분군을 사용한다). 암호학적 프로토콜에서 XTR의 응용은 시큐리티(security)를 손상시키지 않으면서 통신 및 연산 오버헤드에서 실질적인 절약을 가져온다. 따라서 기존에 사용되던 암호 프로토콜을 XTR을 이용해서 무선 환경에 적용할 수가 있다[8].

본 논문에서 구현되어진 프로토콜을 살펴보면 먼저, 사전 초기화 단계에서의 공개키 생성, 전송 및 인증기관으로부터의 인증서 송수신을 전제로 하고, 다음 단계부터는 실제 구매를 위한 지불 단계부터 기술한다. (그림 4)는 전자지갑 기반의 인증 및 지불보안 프로토콜의 흐름을 나타낸다.



(그림 4) 모바일 전자지갑 기반의 지불 보안 프로토콜 흐름도



- m값(g^{Nu}) : 전자지갑의 요청 메시지
- Key_v : 지불서버에서 생성된 세션키
- $Cert_v$: 지불서버의 인증서
- Key_u : 전자지갑의 세션키 정보
- e : 암호문
- N_v : 지불서버에서 생성된 임의의 난수
- IDv : 지불서버의 ID
- N_u : 전자지갑의 임의의 난수
- Info : 전자지갑 인증을 위한 정보
- N, s : 검증에 대한 정보

(그림 5) 전자지갑 기반의 인증 및 지불 보안 프로토콜

먼저 ①부터 ④단계까지 단말기의 모바일 전자지갑과 인증기관 사이에 공개키와 인증서를 교환하게 되고, ⑤부터 ⑩단계에서 그 다음 단계를 위한 연산 부하를 줄이기 위해서 이동 단말기를 사용하지 않을때 필요한 값들을 계산하게 된다. 그리고, ⑪단계부터 실제 상호인증이 이루어지는 단계로 이 단계를 기반으로 지불정보에 대한 처리를 하게된다. 이와 같은 흐름에 맞춰서 실제 실행단계는 (그림 5)와 같이 나타낼 수 있다.

(그림 5)는 단말기의 모바일 전자지갑과 지불서버의 세션 공유키를 이용하여 서로의 정보를 암·복호화하여 상호인증이 일어나는 과정을 나타내고 있다. 기존 경북대 AKA 프로토콜에서 제안되어지고 있는 내용과 같이 서비스 요청 시에 필요한 $m = g^{Nu}$는 실 시행 단계에서의 계산량을 줄이기 위해 사전 계산 단계에서 연산되며, XTR-Signcryption 기반의 인증 및 검증 과정으로 설계되었다. 프로토콜에 대한 세부 실행 순서는 다음과 같다[11].

(1) 지불 요청

전자지갑에서 지불에 대한 요청 메시지로 사전에 계산된 m값(g^{Nu})을 전송한다. 물품을 구매할 경우 구매자의 최종 확인 시에 지불 요청 메시지(m)는 지불서버에게로 전송된다.

(2) 지불서버의 세션키 계산

세션키는 지불 요청 메시지에 의한 전자지갑으로부터 수신된 g^{Nu}와 자신이 생성한 임의의 수 N_v값을 이용해서 계산되어지며, 세션키에 대한 계산과정은 ③번 과정과 같이 이루어진다.

(3) 지불서버의 정보 암호화

(2)단계에서 생성된 세션키를 이용하여 자신의 세션키(Key_v), ID (ID_v),난수(N_v)를 암호화 한다.

(4) 지불서버의 정보 전송

전자지갑으로부터 지불 요청 메시지를 받은 지불서버는 난수(N_v)를 생성하고, 세션키를 이용해 지불서버의 세션키(Key_v), 지불서버의 ID(ID_v), 임의의 수(N_v), 인증서($Cert_v$)를 해쉬한 값을 암호화한 후 그 정보들을 전자지갑에게 전송한다(④번과정).

전송된 정보를 전자지갑이 받으면 지불서버를 인증하고, 지불서버로부터 받은 N_v를 이용해서 암호화 세션에 사용할 세션을 만든다. 이후로는 서로 암호화하여 통신한다.

(5) 전자지갑의 세션키 계산

지불서버로부터 수신된 임의의 수(N_v),와 전자지갑의 임의의 수(N_u)를 이용해서 세션키(Key_u)를 구하면, 세션키에 대한 계산은 ⑥번과 같다.

(6) 지불서버가 보낸 암호화된 정보 복호 및 검증

(5)번에서 계산한 세션키(Key_u)로 지불서버에서 보낸 암호화된 정보인 세션키(Key_v), 지불서버의 ID(ID_v), 임의의 수(N_v), 인증서($Cert_v$)를 복호화 한다. 두 값의 결과가 같으면 세션키의 일치를 뜻한다.

(7) 전자지갑 지불정보의 암호화

전자지갑의 지불 인증을 위한 정보(e, N, s)를 지불서버에 암호화하여 전송하는데, 계산된 세션키(Key = Key_v = Key_u)를 이용해서 정보(Info)를 암호화하여 암호문 e 값을 구하고, 검증에 대한 정보 N, s를 생성한다[11].

전자지갑의 지불 인증을 위한 정보(Info)를 검증하기 위해서 N값과 s값을 계산한다. 사전에 계산된 m값과 암호화된 정보인 e값을 해쉬하여서 N값을 구하고 s값을 구한다. N값과 s값에 대한 계산을 ⑧번과 같다. 지불정보(Info)에 대한 전자지갑의 Signcryption은 (e, N, s)이다[11].

(8) 지불서버에서 전자지갑이 보낸 암호화된 정보의 복호

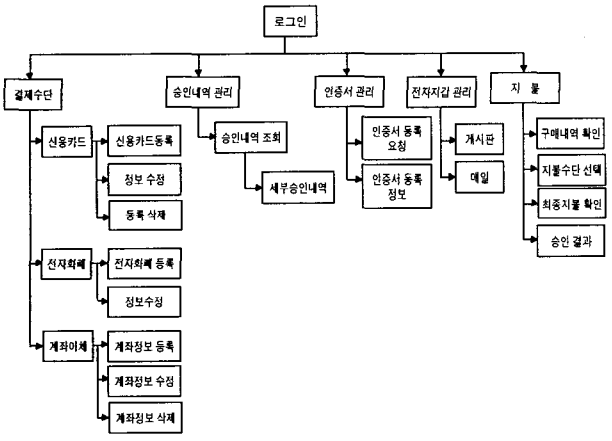
지불서버에서의 복호 모듈은 세션키 Key를 이용해 전자지갑으로부터 전송된 암호화된 정보 e 값을 복호화하고, 복호화된 정보인 값(Info)을 구할 수 있게 된다.

4.3 전자지갑 기능 설계

모바일 전자지갑은 신용카드 정보를 저장하고 있으며 WPKI와 연동하여 사용자가 지불을 원할 경우 데이터를 암호화하여 지불 관련 처리를 하는 어플리케이션으로 다음과 같은 기능을 제공한다(그림 6).

- 지불수단 정보(신용카드번호) 저장기능
- 모바일 전자지갑 접근 관리기능
- 지불서버와의 연동기능
- 상점 서버와의 연동기능
- WPKI와 연동하여 키를 생성하고 인증서 URL을 저장
- 안전한 지불정보 전송을 위한 데이터 암호화/복호화 기능

이러한 기능을 기반으로 기능을 분류하면 (그림 6)과 같다.



(그림 6) 모바일 전자지갑의 기능 분류

모바일 전자지갑은 J2ME 플랫폼 기반의 MIDP 어플리케이션인 MIDlet 형태의 구성되며 사용자는 전자지갑을 다운로드하여 설치, 업그레이드, 실행, 삭제한다. 모바일 전자지갑은 무선 단말기 소유자의 신용카드 정보를 기록하고 저장하는 저장부, 무선단말기 소유자가 전자지갑에 패스워드를 지정하여 다른 사용자가 단말기내의 전자지갑을 구동할 수 없게 제어하는 제어부, 지불 처리시 전자지갑에서 지불 신용카드를 선택하고, 지불을 요청하는 지불 요청부, 무선단말기 소유자가 선택한 지불 수단정보를 전자지불서버에 전송할때 지불정보 암호화 및 WPKI 연동을 위한 암호화 부로 구성된다. 무선 전자상거래를 위한 휴대용 무선 전자지갑의 기능구성은 (그림 7)과 같으며, 각 기능 모듈은 다음과 같다.

(1) 지불처리 모듈 : 신용카드 정보, 사용자정보를 지불서버에 전송

(2) ID, Password 관리 모듈 : 모바일 전자지갑에 사용자의 ID 및 패스워드 등록

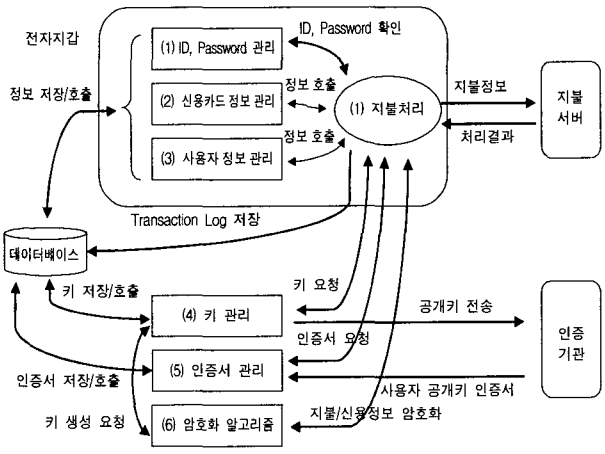
(3) 사용자 정보관리 모듈 : 사용자의 인적사항에 대한 정보를 관리

(4) 신용카드 정보 관리 모듈 : 사용자의 신용카드 정보를 등록, 관리

(5) 공개키 알고리즘 모듈 : 사용자의 공개키/비밀키를 생성

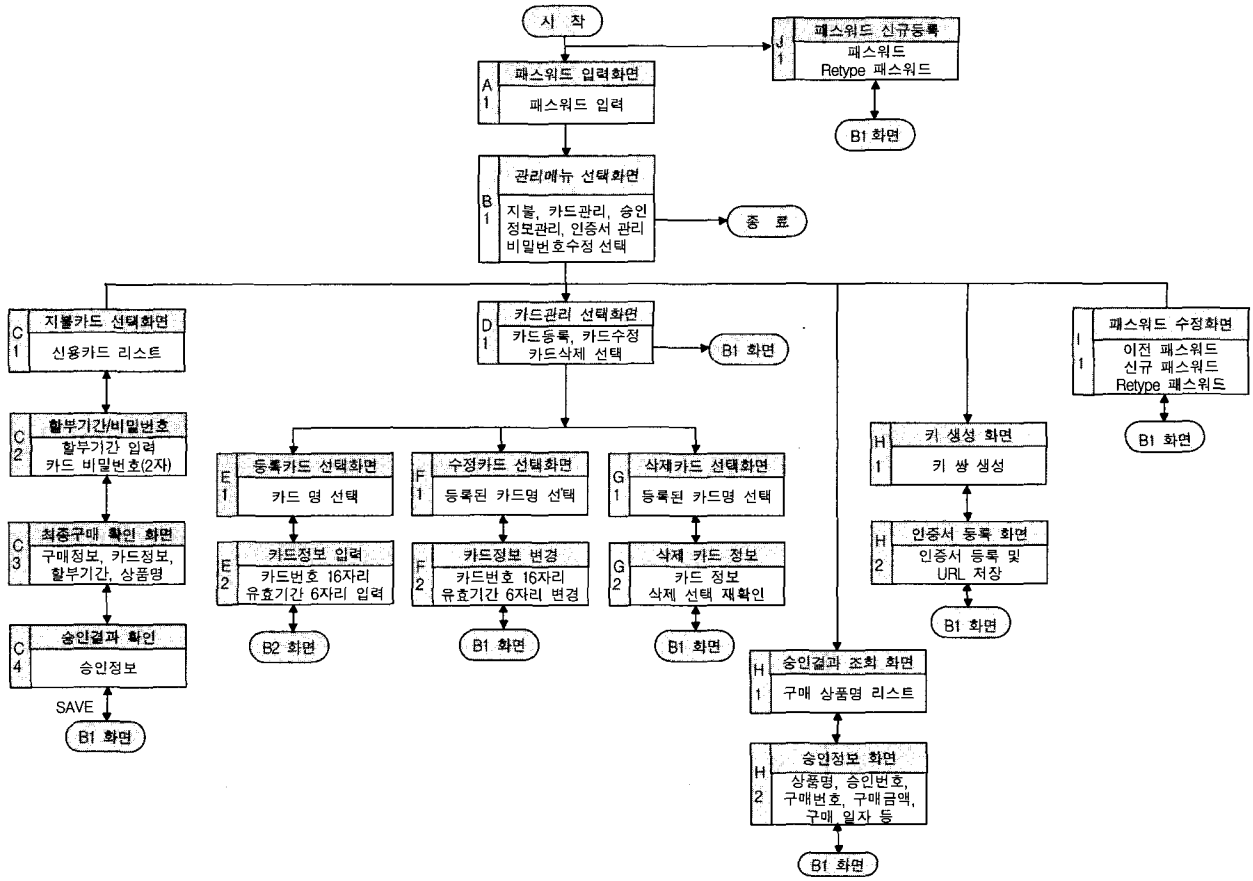
(6) 인증서 관리 모듈 : 인증서 요청 및 URL 저장

(7) 암호 알고리즘모듈 : 지불 정보 암호화



(그림 7) 모바일 전자지갑의 기능모듈 구성도

전자지갑 및 지불서버와의 연동을 고려한 사용자 인터페이스 설계는 (그림 8)과 같다. 상품 구매자가 모바일 전자지갑을 구동하게 되면 단말기의 분실에 따른 개인 정보의 노출을 방지하기 위해 모바일 전자지갑의 아이디와 비밀번호를 입력하게 된다(A1). 모바일 전자지갑이 활성화되면 초기 인증서 발행을 위한 키 쌍을 생성하고 인증서를 등록하고(H1) 인증서 URL을 저장하는 기능과 주문내역을 확인하고 지불을 수행하는 기능(H3), 신용카드 정보를 저장하고(E1) 수정(F1), 삭제하는 기능(G1), 모바일 전자지갑의 비밀번호를 변경하는 기능(I1)을 수행하게 된다.



(그림 8) 모바일 전자지갑 화면 설계도

4.4 전자지갑 모듈 설계

본 장에서는 모바일 전자지갑의 주요 기능모듈인 로그인 모듈, 지불서버와의 통신 모듈, 신용카드, 전자화폐 및 은행계좌 정보 등록 모듈, 금융권 승인내역 관리 모듈의 주요 코드에 대해 설명한다.

(1) 전자지갑 로그인 모듈

```
TextField 아이디 ;
TextField 비밀번호 ;  // 아이디 및 패스워드 필드 생성
new TextField ("아이디", "", 10, TextField.ANY) ;
new TextField ("패스워드", "", 10, TextField.PASSWORD) ;

if (로그인 OK)  // 로그인 결과 처리
   if (구매내역이 존재하면) 결제메뉴 화면으로 전환
   else if (구매내역이 없으면) 관리메뉴 화면으로 이동
else if (로그인 오류) 로그인 오류 메시지 출력

new Connetion(서버 ip, 서버 port) ;  // 로그인 메시지 전송
Connection.sendpacket(아이디, 패스워드, 핸드폰번호 … ) ;
```

지불서버에 접속하여 등록되어 있는 아이디와 패스워드를 확인하여 로그인 후 상품을 구매하고 결제할 내역이 존재하면 지불화면으로 이동하여 지불 진행하며, 지불할 내역이 없으면 전자지갑 관리 화면으로 이동한다.

(2) 지불서버와의 통신 모듈

```
public class Connection   // 클래스 생성
   public Connection (ip, port)
      컨넥션 생성(ip. port) ;
      outputStream 생성 ;
      inputStream 생성 ;

   public sendPacket(데이터)   // 데이터 송수신
      outputStream..writeUTF(데이터) ;  // 데이터 전송 ;
      inputStream.readUTF() ;  // 데이터 수신 ;
```

지불서버로 데이터 전송하는 과정으로 핸드폰에서 지원하는 connection은 한개만 지원하므로 초기 로그인부터 지불까지 한번 이루어진 connection을 활용하여 이루어지게 된다.

(3) 신용카드 정보 등록 모듈

```
Public class AddcardForm extends Form
      TextField 카드번호 ;
      TextField 유효기간 ;

      switch (index)   // 카드 종류 선택
case 0 : "BC" ; break ;
case 1 : "KOOKMIN" ; break ;
case 2 : "KEB" ; break ;   // 입력된 카드의 정보를 저장한다
      RecordStore.데이터저장 (신용카드명, 신용카드번호, 유효기간) ;
```

신용카드 정보를 등록하면 정보는 핸드폰에서 제공하는 DB에 저장하고, 신용카드 비밀번호는 보안을 위해 지불시에 직접 입력하게 된다.

(4) 전자화폐 정보 등록 모듈

```
public class AddcashForm extends Form
   TextField 선불형 전자화폐 발급사 ;
   TextField 전자화폐 번호 ;
// 입력된 카드의 정보를 저장한다
RecordStore.데이터저장(신용카드명, 신용카드번호, 유효기간) ;
```

전자화폐 정보를 등록하면 전자화폐 정보는 핸드폰 내 DB에 저장하고, 전자화폐 지불 비밀번호는 보안을 위해 지불 시에 직접 입력하게 된다.

(5) 은행 계좌 정보 등록 모듈

```
public class AddbankForm extends Form
   TextField 은행명 ;
   TextField 계좌번호 ;
   switch (index) // 은행명 선택
case 0 : 국민은행 ; break ;
case 1 : 기업은행 ; break ;
case 2 : 신한은행 ; break ;
case 3 : 주택은행 ; break ; ……
         // 입력된 계좌 정보를 저장한다
   RecordStore.데이터저장(은행, 계좌번호) ;
```

은행 계좌 정보를 등록하면 계좌 정보는 핸드폰 내 DB에 저장되고, 계좌 비밀번호는 보안을 위해 지불시에 직접 입력하게 된다.

(6) 승인내역 관리 모듈

```
// 선언
   RecordStore ;   // 승인결과 저장 DB
   ChoiceGroup ;   // 승인내역을 초이스 형태로 화면에 표시

// DB에서 승인내역을 불러와서 화면에 나타낸다
   RecordStore.open (승인내역DB, false) ;
   for (레코드 개수)
      ChoiceGroup.append (i번째 승인내역) ;
   RecordStore.close () ;
```

핸드폰 DB에 저장되어 있는 승인완료 내역을 화면에 나타낸다.

(7) 해쉬암호 모듈

```
public class ShaHash extends Hash {
   private void subRound (int count) //. 라운드 함수 {
      int temp = rotateL (A, 5) + f1(B, C, D) +
                 E + w [count] + K1 ;
                 E = D ;
                 D = C ;
                 C = totateL (B, 30) ; // 비선형 함수 적용
                 B = A ;
                 A = temp ; }
      private void subRound4 (int count) {……}
      for (i = 0 ; i < 20 ; ++i )
         subRound1(i) ;        //각 라운드를 20회씩 반복
      for ( ; i < 40 ; ++i )
         subRound2(i) ; ……    // 계속
```

해쉬모듈의 핵심 모듈인 ShaHash에 대한 부문으로 비선형함수와 치환연산으로 이루어진 4개의 라운드 함수를 각각 20회씩 전체적으로 80회 반복 수행한다.

(8) 비 대칭키 암호화 모듈

```
// 개인키 암호화에 대한 "RSA/1/PKCS1Padding"
public void engineSetMode (String mode) throws
NoSuchAlgorithmException {
   if (mode.equals ("1")) {
      this.pkcs1Mode = PRIVATE_KEY_ENCRYPT_MODE ; }
   else if (mode.equals ("2")) {
      this.pkcs1Mode = PUBLIC_KEY_ENCRYPT_MODE ; }
   else {
      throw new NoSuchAlgorithmException ( mode + "Not
      supported" ) ; } }

protected byte[ ] doPadding (int k, byte[ ] data) {
      // PKCS # 1 명세 : EB = 00 || BT || PS || 00 || D
      byte retval[ ] = new byte [k] ;
      retval [0] = (byte)0×00 ;   // 첫 번째 바이트 00
      retval [1] = this.blockType ; // BT

int index = 2 - k + block.length ;
   while (true)   // 데이터의 기본값 계산 {
      if (block [index] == (byte) 0×00) break ;
      index++ ; }
```

5. 모바일 전자지갑의 구현

본 장에서는 J2ME상에서 동작하는 지불보안 프로토콜을 적용한 모바일 전자지갑을 구현에 대해 기술한다. 오프라인에서의 전자지갑 모듈에 기본적인 지불 정보 입력과 모바일 전자지갑을 이용한 유·무선 전자상거래에서의 사용자 인터페이스에 대해 설명한다.

5.1 모바일 전자지갑 구현환경

본 논문에서 구현한 유·무선 연동의 전자지갑은 J2ME 플랫폼을 지원하는 단말기를 이용하여 편리한 사용자 인터페이스를 제공하고 사용자의 개인정보를 관리하는 기능을 구현하였으며, 무선용 공개키 알고리즘을 이용한 보안 모듈을 구현하기 위해 Bouncy Castle 공개키 암호 라이브러리를 이용하여 큰 정수를 처리하는 BigInteger 모듈, 타원곡선 암호 모듈, 공개키 암호모듈을 이용하여 구현하였다[6]. 지불서버는 리눅스 환경에서 자바 언어를 이용하여 개발되었으며, 소켓 프로그래밍을 이용하여 지불 정보를 처리하도록 하였고, 관리자 인터페이스는 자바 언어를 이용한 동적인 서버 사이드 스크립트 언어인 JSP(Java Server Pages)를 이용하여 구현하였다.

5.2 모바일 전자지갑 구현 화면

J2ME 플랫폼을 지원하는 에뮬레이터를 이용하여 구현한 모바일 전자지갑의 화면은 (그림 9)와 같다.

(a) 전자지갑 로그인 화면 (b) 전자지갑 메뉴 화면 (c) 구매정보 확인 화면 (d) 결제 신용카드 선택 화면

(e) 승인결과 화면 (f) 비밀번호 변경화면 (g) 신용카드 등록 화면 (h) 인증서 관리 화면

(그림 9) 모바일 전자지갑의 구현 화면

6. 안전성 및 시험평가

6.1 안전성 분석 및 검증

본 절에서는 논문에서 구현한 프로토콜과 기존 프로토콜을 비교하여 특징과 안전성에 대한 분석을 하였다. 2장에서 설명한 기존 프로토콜과 비교하였으며, 여러 가지 공격 방법들로부터 구현된 프로토콜에 대한 안전성을 설명하였다. 그리고, 검증과정은 서버환경에서 각 모듈에 대해 올바른 결과값이 나오는지를 테스트한 것과 전자지갑 환경에서도 동일하게 구현한 모듈들의 올바른 작동여부를 테스트하였다. 그리고 전체 프로토콜이 올바르게 수행하는지를 검증하였다.

6.1.1 요구사항에 대한 안전성 분석

PACS 프로토콜 방식은 별도의 랜덤 값을 사용하지 않으므로 키 일회성이 없다. 이 방식은 2-Way 방식을 채택함으로서, 사용자의 통신횟수 측면에서 효율성을 높이고 있지만, 최초 정보 전송시 암호화 및 서명을 수행하지 않으므로써 정보의 인증 및 부인봉쇄가 미흡한 단점을 가지고 있다. 또한 사용자가 원하는 통신함수를 선택할 수 없다는 문제점을 갖고 있으며 가입자의 식별자를 그대로 사용하고 있기때문에 익명성을 해칠 수 있는 문제점이 있다.

ASPeCT 프로토콜은 상호 개체 인증, 명확한 키 인증, 키 동의, 키의 신규성에 대한 확인을 사용자와 서버 모두에게 만족시키지만, 익명성과 부인봉쇄는 사용자에게만 만족시키는 단점이 있다.

이동통신 인증프로토콜의 여러 보안 특성중 사용자간의 네트워크 상호 인증 및 익명성의 관점에서 <표 1>에서 주요 프로토콜에 대한 비교 및 프로토콜의 통신횟수와 계산량, 키 생성 방법 등을 비교하였다.

〈표 1〉 요구사항에 대한 안전성 분석 비교

특성 \ 프로토콜	PACS 프로토콜		ASPeCT 프로토콜		구현 프로토콜	
	W→P	P→W	W→P	P→W	W→P	P→W
상호 개체 인증성	△	△	○	○	○	○
함축적 키 인증성	○	○	○	○	○	○
명시적 키 인증성	○	×	○	×	○	×
키 합의	○	○	○	○	○	○
갱신키 확인	○	△	○	○	○	○
이용자의 신분기밀성	△	×	○	○	○	○
상호익명성	○	×	○	×	○	○
부인봉쇄	△	×	○	×	○	○
통신횟수	2		3		3	
암호화	RSA		RSA		DES	
인증키 생성	키쌍 일정		키쌍 임의생성		키쌍 임의생성	
세션키생성	임의선택		임의선택		임의선택	

주) W : Mobile Wallet, P : Payment Server

본 논문의 구현 프로토콜에서 Signcryption은 Gamage, Leiwo, Zheng[7]의 Signcryption을 인증 및 키 교환 프로토콜에 적합하게 변형시켜 사용한 것이다. 구현 프로토콜에서의 Signcryption 사용은 Singnature-then-Encryption 사용시보다 연산량과 통신 횟수 오버헤드를 감소시킨다[11, 12].

6.1.2 구현 프로토콜에 대한 안전성 분석

앞 장에서 설명되어진 요구사항의 내용에 따른 구현 프로토콜의 안전성을 분석하면 <표 2>와 같다.

〈표 2〉 구현 프로토콜에 대한 안전성 분석

구 분	요구사항 만족도 분석	구현 프로토콜에 대한 안전성분석
상호 개체 인증	지불서버가 보낸 임의의 난수에 전자지갑이 서명을 하게 되므로 검증자인 지불서버는 전자지갑을 인증할 수 있다. 또한 전자지갑도 지불서버의 인증서에 기반한 공개키를 이용해 세션키를 생성함으로써 지불서버를 인증할 수 있다.	· 누군가 m을 알아내어 세션키를 계산할 가능성이 있지만, 지불서버의 비밀키를 알지 못하기 때문에 세션키를 계산할 수 없다.
키 인증	전자지갑에서 지불서버로, 지불서버에서 전자지갑으로 모두 명확한 키 인증이 된다. x_U을 아는 사람만이 키를 서명할 수 있으므로 검증자인 지불서버는 세션키가 서명자인 전자지갑으로부터 실제로 계산된 것임을 확인할 수 있다.	· 오래된(이전) 메시지를 사용하여 프로토콜을 진행시키는 Replay Attack의 가능성이 있지만, 프로토콜에서는 임의의 수를 사용하므로 재연 공격을 방지할 수 있다. · 누군가 현재 사용하고 있는 키를 알아내어 이전에 암호화된 메시지를 알아낼 가능성이 있지만, 세션키 생성에 있어서 생성시마다 임의의 수가 변하므로 Forward Secrecy가 보장되어진다.
키 교환 및 키 신규성	키는 지불서버와 전자지갑에 의해 선택된 난수들에 의해서 계산되어진다.	
부인봉쇄 및 익명성	전자지갑에서 지불서버로 가는 사용자 정보가 암호화되어서 전송되므로 익명성이 보장되고, 전자지갑에서 지불서버로 가는 정보에 서명을 하므로 전자지갑은 지불서버에게 보낸 정보를 부인할 수 없게 된다.	· 네트워크측의 내부자가 전자지갑의 정보를 공격자에게 넘겨주어 공격자가 e, N, s를 계산하여 사용자로 가장하여 지불서버에 접근할 가능성이 있다. 그러나 전자지갑의 비밀키를 모르기 때문에 s를 위조할 수 없으므로 프로토콜을 더 이상 진행할 수가 없게 된다.

6.2 성능평가

모바일 전자지갑이 포함된 전자지불시스템의 통합테스트 및 성능 평가는 동시 처리 능력 및 처리시간 평가에 중점을 두어 실행하였다.

- J2ME 기반 트래픽 발생 모듈 생성
- 5대의 컴퓨터에서 J2ME 에뮬레이터를 통해 1000회 트래픽 발생
- 분당 처리능력 및 건당 처리능력 평가

6.2.1 시스템의 구성 환경

설계된 프로토콜을 적용시켜 구현된 시스템의 구성은 자바 환경의 J2SE, J2ME 환경을 기반으로 한다. 지불서버는 J2SE의 java.net 패키지를 이용하여 TCP네트워크를 구현하였으며, 휴대폰 전자지갑에서는 J2ME 환경의 MIDP에서 지원되는 HttpConnection을 이용하였다. 전체시스템의 미들웨어로서 JRUN을 설치하여 웹 서버와 함께 기동할 수 있도록 하였다.

6.2.2 시스템의 시험평가 방법

전체 시스템의 성능평가를 위해서 모듈별 및 전체 프로토콜에 대해서 성능 측정을 실행하였다. 성능 측정 방법은 실행횟수와 입력값에 따라 각 모듈별로 서버와 전자지갑으로 나누어 테스트하고, 각 모듈별 측정값에 따라 성능을 평가하였다. 또한 전체 프로토콜의 실행시간을 특정하여 성능 테스트를 하였다.

지불서버에서의 전체 프로토콜은 소켓을 이용한 TCP 네트워크로 구현된다. 우선, 서버로 사용할 컴퓨터에 서버 프로그램을 수행시키고 클라이언트로 사용할 컴퓨터에서 클라이언트 프로그램을 수행시킨다. 이때 클라이언트는 서버 컴퓨터의 IP와 포트번호 6500번으로 접속을 시도한다. 클라이언트가 서버에 접속하게 되면 인증 프로토콜이 실행되게 된다.

시험 평가는 전체 프로토콜 흐름도에서 클라이언트와 서버, 각각이 처리하는 과정을 출력되게하여 계산되어지는 값들과 주고받는 메시지 값들이 바르게 실행되는지를 비교함으로써 이루어진다. 검증 단계 이후 성능 측정을 위한 각 모듈별, 또한 전체 프로토콜에 대한 평가는 프로그램의 반복실행 횟수와 입력 값의 길이 변화를 통한 프로그램 실행시간 평가에 중점을 두었다.

(3) 시험결과

프로토콜 실행시간은 사용자가 요청메시지를 서버쪽으로 보내는 것부터 최종적으로 서버로부터 인증 성공/실패 메시지를 받는 것까지 측정한다. 각 실행횟수는 1000번의 반복테스트를 통한 평균값을 기록하였다.

〈표 3〉 성능측정 시험결과

구분(문자수)	1	123	1~30	1~100
SHA-1 성능측정	263ms	275ms	277ms	312ms

구분(비트)	32bit	64bit	96bit	128bit	160bit	192bit
DES Encryption 성능측정	6163ms	6201ms	11379ms	12398ms	16532ms	16988ms
DES Decryption 성능측정	6152ms	6199ms	11998ms	12646ms	16424ms	17006ms

구분(함수명)	add()	multiply()	mod()	modpow()	modInverse()
지불서버 측 BigInteger 성능측정	19ms	19ms	16ms	5787ms	1059ms
전자지갑 측 BigInteger 성능측정	19ms	19ms	19ms	2381ms	569ms

성능 테스트 모듈을 통한 실험결과의 데이터를 수집하여 분석한 결과를 보면 시스템의 동시 접속자 처리능력은 시스템에 따라 차이를 보였으며, 시스템의 성능에 따라 가변적이고, 안전한 통신망의 환경을 기반으로 한 것임을 알 수 있다. 또한, 지불 프로토콜 수행 단계에서 사용되는 연산 모듈인 Hash, DES, BigInteger에 대해서만 실행시간을 측정한 결과 SHA의 경우 입력 값이 512bit보다 작을 때는 그 길이에 관계없이 블록의 크기가 512bit로 패딩이 이루어지므로 실행시간은 입력 스트링의 길이에 관계없이 거의 평균적인 속도가 나옴을 알 수 있었으며, DES 암호화는 입력 값을 64bit의 배수로 패딩하여 64bit 블록 단위로 암호화하므로 패딩 과정은 수행 시간에 거의 영향을 미치지 않고 DES 복호화도 DES 암호화처럼 패딩된 비트 제거과정은 실행시간에 거의 영향을 미치지 않았다. 구현한 BigInteger 모듈의 경우 modPow()와 modInverse() 함수를 제외한 함수들은 빠른 연산속도를 나타냄을 알 수 있다. 그러나 modPow()와 modInverse() 함수는 멱승 처리 과정에서 다소 시간이 걸림을 알 수 있다. J2SE에서 제공하는 BigInteger 클래스보다는 연산속도가 느리지만 비슷한 연산처리속도를 제공하였다. 전체 프로토콜의 성능 측정 결과는 <표 4>와 같다. 각 키 길이에 대한 수행시간 측정결과는 1000번의 반복테스트를 통한 평균값을 기록하였다.

〈표 4〉 모바일 전자지갑 성능 측정 결과

키 길이(비트)		10 bit	40 bit	100 bit	170 bit
측정 시간	구현 프로토콜	136ms	752ms	1143ms	1507ms
	PACS 프로토콜	105ms	893ms	1562ms	1982ms
	ASPeCT 프로토콜	159ms	816ms	1357ms	1841ms

PACS 프로토콜 및 ASPeCT프로토콜은 기 구현된 프로토콜에서의 멱승계산량에 따른 비교수치에 의해 결과값을 도출하여 <표 4>와 같이 비교하였다. XTR[8]에서 RSA 1024비트에 해당하는 170비트의 키 길이를 기준으로 테스트 한 결과 전체 지불결제가 실행되는 프로토콜의 시간이 1.5초 정도의 시간이 걸렸다. <표 4>에서 알 수 있는 것처럼 본 논문에서 구현한 프로토콜은 키의 길이가 증가함에 따라 기 구현된 프로토콜에 비해 전자지갑 성능이 향상되고 있음을 나타내고 있다.

7. 결 론

본 논문에서는 무선 전자상거래가 활성화됨에 따라 편리하고 안전하게 지불을 수행할 수 있는 모바일 서비스 플랫폼을 이용한 전자지갑을 구현하였다. 현재 무선 전자상거래를 위한 신용카드 지불을 처리할 수 있는 적합한 방식이 서비스되고 있지 않는 시점에서 모바일 서비스 플랫폼을 이용한 전자지갑 방식은 기존의 인프라를 활용하여 서비스 도입이 가능하다는 점에서 큰 장점이 있다.

무선 전자상거래가 활성화되기 위해서는 보안 문제 해결이 필수적이며 현재 추진되고 있는 WPKI 연동을 통한 사용자 인증 및 신뢰성 있는 거래가 이루어져야 한다. 무선 단말기는 연산속도 및 메모리의 제한으로 유선 환경과는 차별적인 보안 대책이 필요하며 이를 위해 본 논문에서는 알고리즘 수행 속도가 빠르고 작은 키 사이즈로 높은 암호화 강도를 제공하였으며, 지불처리시 전자지갑과 지불 서버간 WPKI 인증서를 이용한 연동방안을 제시하였다. 또한 전자지갑의 편리한 인터페이스 제공을 위하여 현재 게임 등 엔터테인먼트 위주의 서비스를 제공하고 있는 모바일 플랫폼을 이용한 전자지갑 구성 방안을 제시하여 유·무선 전자상거래를 위한 새로운 서비스 플랫폼으로서의 활용이 기대된다.

향후, WPKI 서비스와의 연동이 필요로 하며 모바일 플랫폼에서의 인증서 사용이 가능하도록 단말기의 지원 등 해결해야 할 과제가 많다. 또한 단말기의 연산 능력이 계속 향상되고 있기는 하나 전자지갑에서 완벽한 보안을 제공하기 위해서는 키 생성 및 암호화 속도를 높이기 위한 타원곡선 알고리즘의 개선도 필요하다. 그리고 계좌이체, 전자화폐 및 이동통신사가 서비스 준비중인 스마트카드 기반의 유·무선 전자지불 서비스와의 연동을 위한 지속적인 연구가 필요하다.

참 고 문 헌

[1] A. Mehrotra and L. S. Golding, "Mobility and Security Management in the GSM System and some Proposed Future improvements," Processdings of the IEEE, Vol.86, Issue.7, pp.1480-1497, 1998.

[2] G. Horn and B. Preneel, "Authentication protocols for personal communication systems," Proceedings of ACMSIGCOMM '95, pp.256-261, August, 1995.

[3] Katrina Bond, "Danny Willians, Mobile E-commerce Analysis," Analysis Publication, 2000.

[4] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, "NTRU : A Ring Based Public Key Cryptosystem," Portland, OR, June, 1998.

[5] Jeffrey Hoffstein and Joseph H. Silverman, "[MiniPASS] Authentication and Digital Signatures in a Constrained Environment," Workshop on Cryptographic Hardware and Embedded Systems 2000 (CHES 2000), 2000.

[6] Bouncycastle java library, http://www.bouncycastle.org.

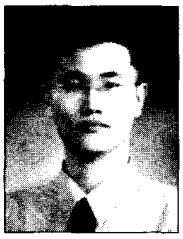
[7] Y. Zheng, "Digital Signcryption or How to Achieve Cost (Signature & Encryption) << Cost(Signature) + Cost(Encryption) >>," Advances in Cryptology, Proceedings of CRYPTO '97, pp.165-179, August, 1997.

[8] Arjen K. Lenstra, Eric R. Verheul, "The XTR public key system," Proceedings of Crypto2000, LNCS 1880, Springer-Verlag, 2000.

[9] CLDC/MIDP, http://java.sun.com.

[10] 이대하, "이동거래를 위한 J2ME기반 전자서명 및 전자지불 시스템 설계", 경북대학교 석사학위논문, December, 2000.

[11] 경북대, "무선 인터넷 기반의 인증 및 키 교환 프로토콜에 관한 연구", 한국전자통신연구원보고서, January, 2002.
[12] NTRU사의 기술문서, http://ww.ntru.com.
[13] JTC(air)/94.12.15-119Rc, "Personal Communications Services," PACS Air interface Specification, PN3418.
[14] Jonathan Knudsen, JAVA Cryptography, O'Reilly, 1998.
[15] 모바일 자바 정보 공유 사이트, http://mobilejava.co.kr.
[16] JMI 암호화 연구소 자바암호 Q&A, http://crypto.jmi.co.kr/question/faq8.html.
[17] J2ME 홈페이지, http://java.sun.com/j2me.
[18] 송유진외 5명, "전자상거래 보안기술", 생능출판사, 1999.
[19] 오카모토, 야마모토 저, 송유진외 1인 번역, "현대암호", 생능출판사, 1999.
[20] 박남제, 신균호, 최영진, 송유진, "M-Commerce를 위한 자바 모바일 플랫폼 기반의 전자지불 구현방안", 2001년 정보처리학회 춘계학술발표논문집, pp.817-820, 2001.
[21] 박남제, 송유진, "M-Commerce Security Platform based on WTLS and J2ME", 2001 IEEE International Symposium on Industrial Electronics, FrM5-06, 2001.
[22] 박남제, 송유진, "모바일 서비스 플랫폼 기반의 무선 전자상거래 보안기술", 2001년 정보보호학회 학회지 8월호, 2001.
[23] 박남제, 최영진, 홍범기외 2명, "M-Commerce를 위한 모바일 자바플랫폼 기반의 보안 메커니즘 설계", 한국통신학회 NCS 2001, December, 2001.
[24] 박남제, 신균호, 홍범기, 송유진, "모바일 서비스 플랫폼 기반 유·무선 연동의 전자지갑 설계 및 구현", 2002년 정보보호학술발표회논문집, pp.313-333, 2002.



박 남 제

e-mail : njpark@mail.skku.ac.kr
2000년 동국대학교 정보산업학과 졸업
2001년~현재 성균관대학교 정보통신대학원 정보보호학과 석사과정
2000년~현재 (주)뉴레카 정보통신연구소 선임연구원
관심분야 : 전자상거래 보안, 무선인터넷 보안, 전자지불시스템, 컨텐츠 보호 등

송 유 진

e-mail : song@mail.dongguk.ac.kr
1982년 한국항공대학교(학사)
1987년 경북대학교(석사)
1995년 일본 Tokyo Institute of Technology (박사)
1988년~1996년 한국전자통신연구원 선임연구원
1996년~현재 동국대학교 정보산업학과 교수
1998년~현재 한국정보보호학회 이사
1998년~현재 ISO/IEC JTC1/SC27-Korea 전문위원
2001년 ICISC2001 운영위원장 역임
관심분야 : 암호 및 인증이론, 전자상거래보안 응용, 전자화폐/전자지불, 무선인터넷 보안, 스마트카드보안, 컨텐츠 보호 등