# 비밀키 안전성을 고려한 인증 프로토콜 설계 및 구현

박 소 희*°, 김 춘 길**, 김 순 자*

* 경북대학교 전자전기공학부

** 한국통신 멀티미디어 연구소

# Design and Implementation of Certification Protocol Considered on Security of Private key

So-Hee Park *° , Chun-Kil Kim** and Soon-Ja Kim*

* School of Electronic and Electrical Engineering, Kyungpook National University

** Korea Telecom Multimedia Research Laboratory

## 요 약

컴퓨터 통신망과 인터넷의 발달로 등장한 전자상거래는 안전성과 신뢰성이 보장되어야 한다. 이를 위해, 본 논문에서는 기존의 인증기관과 전자지갑을 기능별 구조로 고찰한 뒤 비밀키의 안전도를 개선할 수 있는 전자상거래에서의 인증 프로토콜을 제안하고 구현한다. 제안한 인증 프로토콜은 사용자의 공개키/비밀키 쌍을 전자지갑이 직접 생성하고 인증기관은 사용자의 공개키만을 전자서명하도록 함으로써 전자상거래 시스템에서 사용자 비밀키에 더 높은 안전성과 신뢰성을 제공한다. 또한, 공개키 암호시스템으로는 키 길이와 메모리 면에서 효율적인 타원곡선 암호시스템을 사용한다.

## I. 서론

컴퓨터 통신망의 발달과 월드와이드웹(world wide web)의 등장으로 인터넷의 이용이 폭발적으로 증가하였다. 인터넷의 이용이 보편화되면서 여러 가지 서비스가 인터넷에서 제공되고, 이는 기존의 물리적인 시장의 개념과 함께 가상 공간에서의 전자적인 거래의 개념도 도입하였다.

전자상거래(electronic commerce, EC)는 인터넷이라는 불안전한 개방형 전산망에서 거래가 이루어지므로 일반 사용자에게 널리 보급되기 위해서는 기존의 시장이 가진 특성과 더불어 개인 정보 노출 및 변조의 위험성을 최소화하여 거래의 안전성과 신뢰성을 보장할 수 있어야 한다. 즉, 전자상거래 사용자의 신분 확인과 인증이 필수적이며 거래의 내용과 개인 정보의 보호가 요구된다. 이들을 제공하기 위해서는 기본적인 암호학적 정보보호 기술의 도입이 필요하며, 특히 사용자의 인증과 거래의 신뢰성을 보장하는 디지털 서명 기술 등에 사용되는 공개키 암호시스템의 적용이 필요하다.

공개키 암호시스템은 대칭키 암호시스템의 비밀키 보관 문제와 키 분배 문제를 해결하였으나, 공개된 키가 원하는 사용자의 공개키가 맞는지를 확인하는 수단이 필요하다[1]. 이를 해결할 수 있는 것이 인증기관으로 사용자의 공개된 키와 이와 관련된 사용자 정보의 인증을 통하여 전자상거래의 안전성과 신뢰성을 보장할 수 있다.

또한, 안전한 전자상거래를 위해서는 인증기관과 더불어 사용자의 공개키/비밀키 쌍을 직접 생성하고, 비밀키를 안전하게 보관해 줄 수 있는 전자지갑이 필요하다. 전자지갑은 사용자의 키 쌍 생성과 비밀키, 개인 정보 및 확인서를 안전하게 보관하며, 전자상거래 시 안전한 지불과 관련된다.

기존의 VeriSign[2]과 같은 인증기관에서는 확인서 발급 시에 인증기관이 직접 사용자의 공개키와 비밀키를 생성하여 비밀키를 사용자에게 전달하고 자신이 소유하고 있는 비밀키는 파기하는 형태를 취하고 있다. 이는 인증기관이 사용자의 비밀키를 알고 있어 사용자가 비밀키를 분실했을 경우 복구는 편리하나, 전자상거래와 같이 고객의 개인 정보 보호가 요구되는 경우에는 인증기관의 비밀키 보관은 안전성과 신뢰성에 문제가 될 수 있다. 그리고, Java Wallet[3] 및 CyberCash Wallet[4]과 같은 전자지갑은 확인서를 전자지갑에 직접 보관하지 않도록 되어 있어 지불 시에 안전성과 효율성을 제공하기에 충분하지 않다. 이와 같이 인증기관과 전자지갑은 연동을 고려하지 않은 경우는 전자상거래에서 요구되는 안전성과 효율성을 사용자에게 제공하기에 충분하지 않으므로, 전자상거래에서 사용하기에 적합하도록 두 기관의 연동을 고려한 구현이 필요하다.

본 논문에서는 기존의 인증기관과 전자지갑을 기능별 구조로 고찰한 뒤 전자상거래에서 요구되는 비밀키의 안전도를 제공할 수 있는 전자지갑을 이용한 인증 프로토콜을 제안하고 구현하며, 이에 타원곡선 암호시스템을 적용한다. 제안한 인증 프로토콜은 사용자의 공개키/비밀키 쌍을 전자지갑이 직접 생성하고 인증기관은 사용자의 공개키만을 전자서명하도록 함으로써 전자상거래 시스템에서 사용자 비밀키에 더 높은 안전성과 신뢰성을 제공한다.

## II. 인증기관

전자상거래가 실현되기 위해서는 전자지갑을 장착한 고객시스템, 인증기관, 상점시스템 및 지불시스템이 존재해야 한다. 인증기관은 이러한 전자상거래에 참여하는 개인 및 기관을 확인하고 인증해 주기 위해 확인서(certificate)를 발행, 개정, 취소하는 기관으로서 실제로는 거래 당사자가 사용하는 공개키를 인증해 준다. 확인서는 해당 요소시스템이 전자상거래 시 사용하게 될 공개키와 관련된 정보들로 구성되어 있다. 이러한 역할로 인해 인증기관은 전자상거래의 핵심 부분이라 할 수 있다.

일반적인 확인서 발급 절차는 우선 사용자가 자신의 공개키/비밀키 쌍을 생성하여 비밀키는 자신이 보관하고 공개키는 실명확인이 가능한 인증 정보와 함께 인증기관에게 보내어 확인서 발급을 요청한다. 그러나, 기존의 인증기관은 사용자 비밀키의 복구 편이를 위해 인증기관이 직접 사용자의 공개키/비밀키 쌍을 생성하고, 보관하도록 되어 있다. 이 경우는 인증기관이 사용자의 비밀키를 알고 있으므로 비밀키에 대한 충분한 안전도를 제공하지 못할 수 있다. 확인서 발급기관은 전달된 인증 정보를 확인서 승인기관을 통해 확인하여 사실임이 증명되면 확인서를 발급한다. 인증기관의 운영과 관련하여 여러 가지 사항들이 고려되어야 하는데, 이러한 고려사항을 문서화해 놓은 것을 CPS(certificate practice statement)라 한다. 특히, 인증기관의 트러스트 체인, 사용 알고리듬과 키 길이, 확인서 용도제한 등은 중요한 정책 중 하나이다.

근본적으로 이러한 인증기관은 공개키 기반 구조(public key infrastructure, PKI)를 지원하기 위한 것이며, PKI 는 공개키 암호 시스템을 활용하여 개인의 공개키를 신뢰할 만한 기관에서 관리하는 구조로서 현재 미국, 캐나다 등 일부 선진국과 산업체 및 표준화 단체에서 연구, 개발되었거나 개발 중에 있다[5].

### 1. FPKI

FPKI 는 미국의 연방정부에서 안전한 전자거래와 전자문서의 인증, 공개키 확인서 관리, 보안 서비스 등을 위해 만든 공개키 기반 구조이다[6-9]. 이러한 PKI 를 통하여 사용자는 특정한 구조나 구현에 관계없이 전자상거래에 참여하고 안전하게 통신할 수 있어야 하며, 확인서의 생성 및 취소를 요구할 수 있어야 한다. 또한 원하는 상대방의 공개키를 확인하기 위해 상대방의 확인서를 획득하고 해석, 확인할 수 있어야 한다. 이러한 요구 조건을 만족하기 위해 FPKI 는 다음과 같은 구조의 공개키 기반 구조로 구성 운용된다.

FPKI 는 정책승인기관(policy approving authority, PAA), 인증기관(certificate authorities, CAs),

등록기관(organizational registration authorities, ORAs) , PKI clients, 디렉토리 서버(directory server, DS) 등으로 구성된다.

PAA 는 정책승인기관으로 최상위의 root CA 와 직접 연계되어 하위 기관에 대한 책임을 규정하고 위임한다. 즉, CA 들의 운영 정책과 확인서 발급 정책을 관리하고 모니터한다.

CA 는 인증에 관련된 일련의 동작을 수행하는데 확인서를 생성, 발급, 보관 및 취소하는 역할을 한다. 그러므로 확인서를 보관하고 있는 디렉토리 서버와 연계되어 동작해야 하며 명확한 CA 운영 정책이 필요하다. 상위 CA 는 하위 CA 의 정책을 제한할 수 있다.

ORA 는 CA 에게 사용자의 확인서 생성을 요청하는 기관이다. 사용자가 ORA 에게 확인서 발급을 요구하면 ORA 는 사용자와 사용자의 공개키가 적법한지를 확인하고, CA 에게 확인서 생성 및 발급을 요구한다. 확인서가 CA 에 의해 생성되면 사용자의 공개키 확인서를 소유자에게 전달한다.

PKI client 는 일반적으로 사용자를 말한다. PKI 에서는 사용자가 직접 자신의 키 쌍을 생성하고, 이 중 자신의 비밀키를 이용하여 전자서명을 하고, 상대방의 공개키를 이용하여 전자서명을 확인한다.

DS 는 디렉토리 서비스를 의미한다. 사용자가 온라인으로 접속 가능하며, 모든 CA 는 이 DS 를 가지고 있거나, RFC 1487 에 정의되어 있는 LDAP(lightweight data access protocol)을 통해 DS 에 접근 가능하다[10]. 또한, DS 는 X.500 디렉토리[11]를 사용하며 미국의 디지털 서명 표준인 DSA[12]를 사용한다.

상기한 구성 요소들로부터 PKI 는 각 사용자의 공개키 확인서를 발급하고 확인서에 대한 인증을 제공한다. 사용자가 인증을 확인하는 경로로는 계층적 구조와 망 구조가 있다. 일반적으로 두 가지 방법이 모두 제공되며 두 가지를 혼합된 형태를 많이 이용된다.

FPKI 에서 각 사용자의 확인서는 ITU-T 의 X.509 권고안의 v3 형태를 따르고, 취소 확인서 리스트(certificate revocation list, CRL)는 X.509 의 v2 를 따른다[13].

2. VeriSign

1995 년 RSA 사에서 설립한 회사로 디지털 인증 서비스와 안전한 통신을 위해 digital ID 라는 인증 서비스를 3 개의 class 로 구분하여 제공하고 있다[2]. class1 은 개인의 이름이나 전자우편주소 등을 인증해 주는 서비스이고, class2 는 공적인 정확도를 갖는 개인정보를 전제로 개인의 이름, 주소 등을 인증해 주며 전자상거래 소비자 입장에서는 보통 수준의 인증 서비스이다. class3 는 인증 레벨을 높여 은행구좌 개설, 계약서 작성, 소속기업의 증명 등을 인증해주며 기업 및 단체에게도 확인서를 발급한다. 그러나, 현재 구체적인 서비스 사양과 범위 등은 미정이다.

VeriSign 은 서로 다른 IA(issuing authority)로 구성된 PKI-entity 계층 내에서 구현되는데 IA 는 CPS 의 범위 내에서 확인서를 발행, 중지 및 취소하는 VR(VeriSign root), PCA(primary certification authority), CA(certification authority)를 통틀어서 의미한다. VeriSign PKI 는 VR, 셋 이상의 PCA, 셋 이상의 CA 와 인증된 IA 에 의해 인증된 다른 CA 로 구성된다.

VR 은 root 로서 VeriSign 에서 최고의 신뢰기관이며 PCA 공개키를 위한 확인서를 발행한다. PCA 는 모든 CA 를 위한 확인서 발행, 중지와 취소를 담당하며, 적법한 경우 non-VeriSign PKI 의 대응되는 entity 와 상호 인증을 할 수도 있다. CA 는 한 PCA 의 하위로서 CPS 와 PCA 에 의해 규정된 규약에 따라 운영되며, 사용자의 확인서와 관련된 일련의 동작들을 수행한다.

3. SET(secure electronic transaction)

비자카드와 마스터카드가 안전한 전자상거래를 위해 공동 개발한 프로토콜로써 인터넷을 통한 안전한 신용카드 거래시스템 구현을 목적으로 한다[14].

SET 은 카드를 사용하는 고객, 카드 발행사, 상점, 전표 매입사, 인터넷 지불 게이트웨이(payment gateway), 카드사, 인증기관 등으로 구성되어 있으며 안전한 전자상거래를 위해 다음과 같은 조건을 만족하도록 한다. 첫째, 상품주문정보와 대금지불정보를 보호해야 하고, 전송된 자료의 무결성을 보장해야 한다. 둘째, 카드사용자가 카드회사 계좌의 합법적인 사용자임을 인증해야 하고 상점이 카드회사와 거래할 수 있음을 인증해야 한다. 셋째, 거래를 부인할 수 없도록 전자영수증이 발급되어야 한다. 이러한 조건을 만족하기 위해 SET 에서는 RSA 서명을 이용하여 전자봉투, 이중서명과 같은 암호학적인 보호방법을 사용하고 있다[15]. 그러나, RSA 와 같은 공개키 암호시스템을 사용하기 위해서는 사용자와 키의 정당성을 확인할 수 있는 인증기관이 필요하다. SET 에서는 계층 구조로 각각의 시스템에게 확인서를 발급한다. 먼저 root CA 가 카드 brand CA 에게 확인서를 발급하고 카드 brand CA 는 지방의 CA 에게 확인서를 발급한다. 각 지방의 CA 는 카드소유자 CA, 가맹점 CA 및 payment CA 를 각각 확인해 준다. 그러면 사용자인 카드소유자, 상점 및 지불게이트웨이는 자신이 속한 CA 에게 확인서를 발급받는다.

확인서 포맷은 FPKI 와 마찬가지로 ITU-T X.509 v3 를 따르고 서명 알고리듬은 RSA 를 사용한다. 또한 CRL 은 ITU-T X.509 v2 의 포맷을 따른다.

## III. 전자지갑

안전한 전자상거래가 실현되기 위해서는 2 장에서 살펴본 바와 같이 공개키 암호시스템의 적용이 필수적이다. 공개키 암호시스템에서 사용자의 비밀키가 노출될 경우 원하지 않는 사용자가 암호문을 복호하거나 합법하지 않는 디지털 서명이 생성될 수 있다. 그러므로 사용자의 비밀키는 안전하게 보관되어져야 한다. 이를 해결할 수 있는 것이 전자지갑으로 사용자의 PC에 장착된 전자지갑은 사용자의 공개키/비밀키를 생성할 뿐만 아니라 생성된 비밀키와 공개키의 확인서를 안전하게 보관하며 전자서명 생성/확인 등의 암호학적 기능을 수행한다. 또한 지불처리 시에 이중서명(dual signature)을 사용하여 안전한 지불을 가능하게 한다. 이러한 전자지갑이 전자상거래에 사용되기 위해서는 사전에 인증기관으로부터 전자지갑의 소유자에 관한 정보와 원하는 지불수단에 대해 확인서를 발급받아야 하며, 지불 수단으로는 신용카드(credit card), 직불카드(debit card), 자금이체(fund transfer), 전자화폐(electronic cash) 등이 이용될 수 있다.

현재 전자지갑은 지불 수단별로 PC 장착형 소프트웨어로 구현되고 있으며, 그 유형에는 크게 plug-in 형, helper 형, proxy 형이 있다[16]. 이들 방법 이외에도 SHTTP(secure hypertext transfer protocol)[17]을 이용하거나 HTTP 의 PEP(protocol extension protocol)[18]를 이용하여 전자지갑을 구현할 수 있다. 앞으로 스마트 카드와 같은 chip card 가 보편화될 경우에는 이를 이용한 통합 전자지갑으로 발전될 것이다.

1. Java Wallet

안전한 전자상거래 실현을 위해 클라이언트 측면의 프레임워크를 제공하는 썬 마이크로시스템의 제품으로 commerce JavaBeans 와 Java commerce client 를 포함하고 있다[3]. commerce JavaBeans 는 암호와 상거래에 관련된 인터페이스를 지원하며 모듈별로 재사용 가능한 commerce component 를 제공하는 JavaBeans specification 의 확장이며, Java commerce client 는 안전한 전자상거래 구성 요소의 상호 작용과 동적인 다운로드 및 인스톨 등을 가능하게 해 주는 commerce Bean box 이다.

Java Wallet 은 브라우저에서 plug-in 처럼 인스톨되는 Java Activator 와 호환되어 "write once, run anywhere"이 가능하며, 모듈별 전자상거래 구성 요소를 제공하기를 원하는 상거래 개발자들과 온라인 상거래를 수행하는 데 있어 사용하기 쉽고, 확장적인 메커니즘을 원하는 client 를 위해 개발되었다.

이러한 Java Wallet 은 고객 시스템 내에 인스톨되며, 사용자는 commerce JavaBeans 구성 요소를 다운로드함으로써 쉽게 기능을 확장할 수 있다. Java Wallet 은 JCM(Java commerce messages)을 이용하여 commerce server 와 상호 운용된다. JCM 은 구매, 자동이체, 현금관리

등의 동작을 제한하기 위한 property list 이다.

Java Wallet 의 사용자들은 호환성있는 지불수단과 프로토콜을 선택할 수 있으며, 주소록과 같은 개인적인 서비스도 억세스할 수 있다. 또한, commerce server 는 SET 과 같은 안전한 프로토콜을 지정할 수 있도록 하여 안전한 거래를 보장하고 있다. 이러한 사용될 수 있는 지불도구와 프로토콜에 관한 정보는 JCM 에 의해 제공되어진다.

Java Wallet 의 안전도는 gateway security 모델이 관리한다. 또한, 안전한 로그인과 신용카드 정보, 주소 등을 저장하기 위한 지역적인 안전한 데이터베이스도 제공하고 있다. 그러나, 실제적으로 SHTTP 나 암호화 알고리듬을 이용한 좀 더 근본적인 안전성은 아직 제공하지 않고 있다.

2. CyberCash Wallet

CyberCash Wallet 은 사용자들이 인터넷 상점으로부터 상품과 서비스를 안전하게 구매할 수 있도록 해 주는 안전한 CyberCash 인터넷 지불 시스템의 중요한 구성 요소로 인터넷에서의 개방된 표준으로 개발되었다[4]. 전자화폐의 일종인 CyberCash 를 위해 개발된 제품이기는 하나 완전한 호환성을 가지고 있어 전자화폐의 종류와는 무관하게 하나의 전자지갑만을 사용하면 된다.

CyberCash Wallet 은 안전한 거래를 위하여 암호화 기술을 이용한다. 즉, 사용자의 신용카드 정보와 구매기록 등은 사용자의 컴퓨터 상에서 암호화 기술을 이용하여 암호화되므로, 사용자의 전자지갑이 lock 만 되어 있다면 사용자의 신용정보 유출은 걱정하지 않아도 된다. CyberCash Wallet 에서는 1024 비트 RSA 와 56 비트 DES[19] 암호시스템을 사용하며, 현재 미국 내에서만 사용하도록 제한하고 있다. 또한, 사용자의 전자지갑을 안전하게 보호하기 위해 wallet ID 를 사용한다. 전자지갑의 wallet ID 생성 시에 사용자의 컴퓨터에 저장되어 있는 사용자의 정보들이 사용자의 신분 확인을 위해 사용된다. 신용카드와 Wallet ID 가 링크될 때 입력되는 정보, 즉 신용카드 번호와 이름 등은 전자지갑 내에서 안전하게 저장되어 사용자가 구매 시마다 카드번호를 입력할 필요가 없도록 하며 구매하는 동안 신용카드를 인증해 주는 역할도 한다. 그러나, 안전한 전자상거래를 위해 필요한 확인서와 이와 관련된 기능은 제공하지 않는다.

3. 국내 개발된 전자지갑

국내에도 여러 업체들이 다양한 전자지갑을 개발, 자사의 지불 서버를 이용하는 홈쇼핑 업체들을 대상으로 제공하고 있다. 데이콤은 자체 개발한 전자지갑 프로그램을 데이콤 내 EC 호스팅 업체들을 대상으로 제공하고 있고, 최근에는 커머스넷코리아를 주축으로 "아이

캐시(ICash)"라는 전자지갑을 개발, 보급 중이다[20]. 이 전자지갑은 신용카드결제와 은행 계좌이체를 함께 지원하고 있으며 IC 카드를 지원하는 골드형과 네트워크만을 지원하는 실버형 두 가지가 있다. 인터넷 보안 전문업체인 이니텍은 2048 비트 RSA 를 사용하는 전자지갑 "이니텍페이"를 개발하였다[21]. 이 외에 메타랜드가 SET 방식을 지원하는 전자지갑[22]을 내놓고 있으며 LG 인터넷도 non-SET 방식의 전자지갑 "넷크레딧"[23]을 선보였다. 이러한 국내 개발된 전자지갑은 대부분 신용카드 지불 방식을 기반으로 하는 전자지불시스템의 일부분으로써 개발되었으며, 인증기관의 확인서와 관련하여 온라인 확인서 발급 기능은 아직 충분한 연구가 이루어지지 않고 있다.

## IV. 제안한 인증 프로토콜

인증시스템은 정책기관(policy authority, PA), 인증기관(certificate authority, CA), 사용자등록을 하는 등록기관(registration authority, RA), 확인서를 저장하는 디렉토리(directory), 및 사용자(user)로 구성된다.

PA 는 CA 의 확인서 발급 기준 및 운용 지침 등을 규정하는 상위기관으로 CA 의 운영정책을 관리한다. 즉, 정책을 수립하고 이 정책을 CPS 로 변환한다. CA 는 합당한 사용자에 대해 공개키를 확인하고 확인서를 발급, 취소, 만료 및 갱신하는 역할을 수행한다. RA 는 사용자가 자신의 공개키에 대한 확인서를 발급받기 위해 신청을 하는 곳으로 사용자의 신분확인을 수행한다. 신뢰도에 따라 온라인 RA 와 오프라인 RA 로 구분된다. 온라인 RA 는 낮은 신뢰 등급을 원하는 사용자에게 온라인으로 용이하게 확인서 신청을 제공하며 오프라인 RA 는 높은 신뢰도를 원하는 사용자에게 face-to-face 를 통한 신분확인으로 확인서 신청을 받는다. 때에 따라 한 RA 가 온라인과 오프라인 등록 모두를 제공할 수도 있다. 디렉토리는 사용자가 확인서를 사용할 수 있도록 확인서와 관련된 정보를 보관하는 공개된 보관소이다. 사용자가 확인서를 사용하고 싶을 때, 디렉토리에 접근해서 필요한 공개키를 얻을 수 있다. 사용자는 소비자, 상점, 지불 게이트웨이 등이 해당된다. 하나의 CA 를 두는 것보다 여러 개의 CA 를 계층적으로 두는 것이 효율적일 경우에는 계층적으로 CA 를 구성할 수 있다.

그러나, 전자상거래에서 요구되는 사용자 비밀키의 안전도를 제공하기 위해서는 인증기관과 전자지갑의 연동이 필요하다. 기존의 인증기관과 전자지갑은 이를 충분히 고려하지 않아 비밀키의 안전성과 신뢰성 및 확인서의 보관 문제 등이 완전하게 해결되지 않았다. 이러한 문제를 해결하기 위해 그림 1 과 같은 전자지갑과의 연동을 고려한 인증 프로토콜

을 제안한다.

사용자는 자신의 PC에 내장된 전자지갑을 이용하여 전자상거래에서 사용하게 될 공개키/비밀키 쌍을 생성한다. 비밀키는 전자지갑 내에 안전하게 보관하고 공개키만을 인증기관에게 전달하여 확인서를 발급받으며, 확인서는 브라우저가 아닌 전자지갑에 보관한다. 인증기관은 브라우저를 통해 사용자가 공개키의 확인서 발급을 요청해 오면, 적법한 사용자인지 신분을 확인하고 공개키 확인서를 발급한다. 발급된 확인서는 브라우저를 통하여 사용자에게 확인을 받고 사용자의 전자지갑으로 전달되며 동시에 디렉토리에 저장된다.

이러한 인증기관을 사용하면 전자지갑이 직접 공개키/비밀키 쌍을 생성하므로 기존의 인증기관이 가진 비밀키의 안전도보다 더 높은 안전도를 제공할 수 있다. 또한, 전자지갑에 확인서가 보관되어 있으므로 지불처리 시에 지불 프로토콜이 전자지갑에 접속하기만 하면 사용자의 정보와 확인서를 동시에 제공받을 수 있어 효율적으로 지불을 처리할 수 있으며, 스마트 카드를 이용한 통합 지갑으로 발전할 경우에도 적용가능하다.

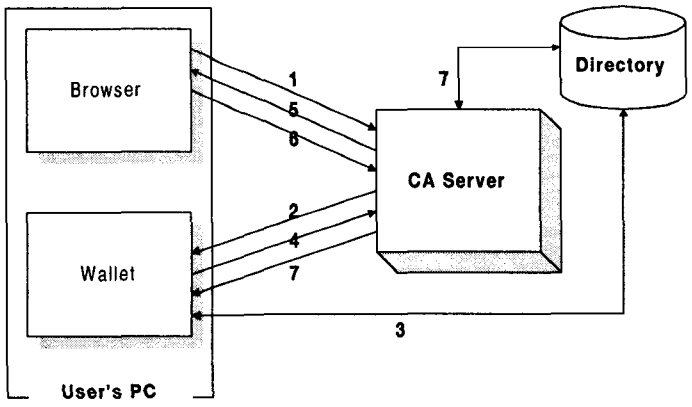


그림 1. 전자지갑과 연동한 인증기관

공개키 암호시스템으로는 타원곡선 암호시스템을 이용한 DSA를 사용하고, 해쉬 알고리듬으로는 SHA1[24]을 이용한다. 타원곡선 암호시스템의 사용은 시스템의 효율성 및 스마트 카드와의 통합을 고려하였다[25].

제안한 인증기관의 확인서 발급 과정은 그림 1에 나타낸 바와 같으며 구체적으로 살펴보면 다음과 같다.

(1) 사용자는 자신의 브라우저로 인증기관 홈페이지에서 확인서 발급을 요청한다. 이 때, 사용자의 전자지갑에서 사용하는 사용자 ID와 패스워드를 입력한다.

(2) 인증기관은 사용자가 입력한 전자지갑의 ID 와 패스워드를 확인한다.
(3) 사용자는 디렉토리에서 인증기관의 확인서를 획득한다.
(4) 사용자는 인증기관의 확인서로부터 인증기관의 공개키를 얻고 이를 이용하여 사용자 정보 및 공개키를 암호화하여 전송한다.
(5) 인증기관은 자신의 비밀키를 사용하여 사용자의 정보와 공개키를 복호하고 (2) 단계에서 확인이 성공했다면 확인서를 발행하고 브라우저 상으로 보여준다.
(6) 사용자는 브라우저 상에 있는 확인서가 자신이 요구한 것이 맞는지 확인하고 이상이 없다면 확인서 발급을 승인한다.
(7) 인증기관은 승인된 확인서를 사용자의 전자지갑에 전송하고 디렉토리에 보관한다.

## V. 구현

### 1. 타원곡선 암호시스템

공개키 암호시스템은 기존의 대칭키 암호시스템이 가지는 비밀키 보관 문제와 키 분배 문제를 해결했을 뿐만 아니라, 디지털 서명의 개념을 등장시켰다. 또한, 다양한 종류의 프로토콜 즉, 전자화폐, 사용자 인증 프로토콜 등이 공개키 암호시스템을 기반으로 설계되었다. 그러나, 공개키 암호시스템이 가지는 여러 가지 장점에도 불구하고 지나치게 큰 키 길이와 암호화/복호화에 소요되는 긴 시간으로 인해 스마트 카드처럼 아주 작은 용량의 컴퓨팅 파워와 제한된 양의 메모리를 갖는 디바이스에는 적합하지 않다.

이러한 공개키 암호시스템의 문제점을 개선할 수 있는 것이 N. Koblitz[26]와 V.S. Miller[27]에 의해 고안된 타원곡선을 이용한 공개키 암호시스템이다. 타원곡선 암호시스템은 타원곡선 위에서 그룹을 정의하고 이에 대한 이산대수문제를 정의함으로써 일반적인 그룹에서 정의되는 이산대수문제보다 해독하기가 더욱 어렵다. 이에 따라, 키 길이와 계산량의 문제를 개선할 수 있게 되었다.

타원곡선 위에서의 이산대수문제(elliptic curve discrete logarithm problem, ECDLP)의 어려움은 $GF(p)$에서 정의된 타원곡선 $E$와 두 점 $P,Q \in E$가 주어졌을 때, $Q=xP$를 만족하는 $x$를 찾는 것이 어렵다는 데 기초한다. ECDLP를 푸는데는 현재까지 Shanks 알고리듬, Pohlig-Hellman 알고리듬 및 Pollard Rho 알고리듬 등이 있으며, 군의 위수가 약 40 자리수(130 비트) 이상인 소인수를 가질 경우 이들 알고리듬에 대해 안전한 것으로 알려져 있다[28].

1024 비트의 소수 p를 사용하는 RSA 시스템과 비슷한 안전도를 가지기 위해서는 ECDLP 에서의 $F_p$나 $F_{2^m}$에서 160 비트 정도의 p나 m을 사용하면 된다[28]. 특히, $GF(2^m)$에서의 타원곡선 암호시스템은 일반적인 필드에서의 연산보다 소프트웨어 및 하드웨어 구현 측면에서 단순하여 많이 이용된다. 유한체 상에서 구현되는 이산대수문제에 기반한 대부분의 암호시스템들은 타원곡선 위에서 암호시스템으로 구현 가능하며, 키 생성, 암호화/복호화 및 디지털 서명에 이용할 수 있다[5].

2. 타원곡선 암호시스템을 이용한 인증 프로토콜

제안한 전자지갑과의 연동을 고려한 인증기관은 시스템간의 이식성을 위해 자바로 구현하며 JDK 1.1을 이용한다. 확인서 발급에 필요한 인증기관 서버는 워크스테이션상에서 구현되며 전자지갑은 펜티엄 PC 상에서 구현한다. 키 생성과 암호화/복호화 및 서명생성/검증은 모두 타원곡선 암호시스템을 이용한다. 타원곡선 암호시스템은 위에서 살펴본 바와 같이 다른 공개키 암호시스템보다 키 길이와 계산량, 메모리 측면에서 효율적이므로 이를 이용한다[25]. 시스템간의 이식성을 위해 타원곡선 암호시스템도 자바로 구현한다. 타원곡선 암호시스템은 $GF(2^{255})$상의 타원곡선 $y^2+xy=x^3+161$을 이용한다. $GF(2^{255})$상에서 구현되므로 1024 비트의 소수 p를 사용하는 RSA 보다 높은 안전도를 얻을 수 있다[28].

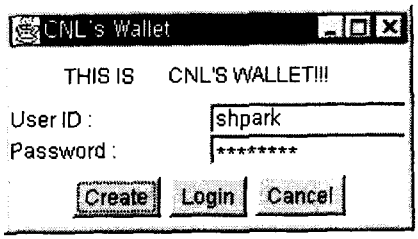


그림 2. 로그인 패널

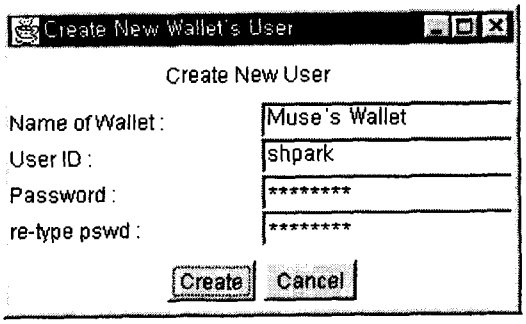


그림 3. 사용자 등록 패널

먼저 확인서 발급을 요청하기 전에 사용자는 자신의 전자지갑에 등록되어 있어야 한다. 그림 2는 전자지갑의 초기 화면으로 이미 사용자 등록이 되어 있다면 user ID와 password

를 이용하여 로그인하고 user ID 가 존재하지 않다면 create 버튼을 이용하여 사용자 등록을 하여야 한다.

사용자 등록은 그림 3 의 패널을 통해 이루어지며 그림 4 에서는 확인서 발급과 지불처리 시에 이용하게 될 사용자 정보를 입력하는 패널이다.

사용자 정보를 모두 입력하고 ok 를 누르면 그림 5 의 키 생성을 위한 다이얼로그 박스가 나타나고 이 때 ok 를 누르면 공개키와 비밀키가 생성된다.

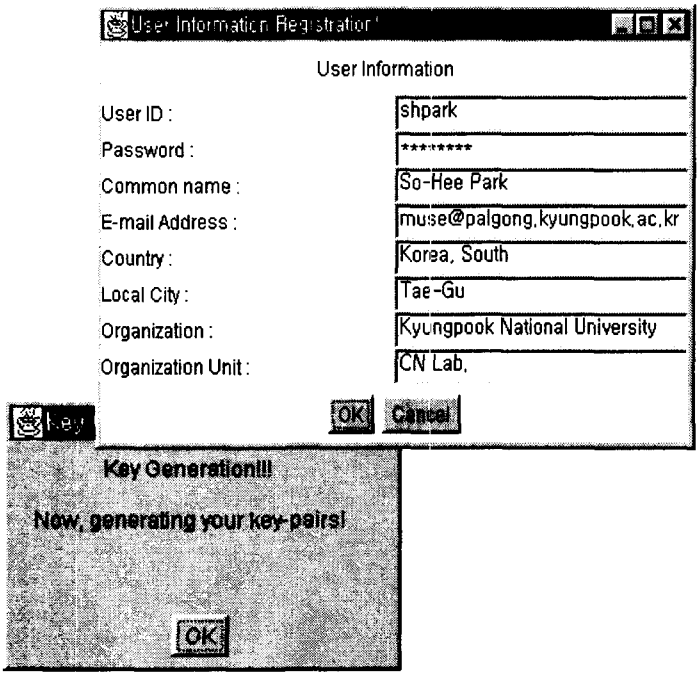


그림 4. 사용자 정보 입력 패널

그림 5. 키 생성 패널

전자지갑에서의 사용자 등록이 완료되면, 그림 6 의 인증기관 홈페이지에서 확인서 발급을 요청한다. 인증기관에서 사용자 공개키에 관한 확인서가 발급되면 그림 7 에서와 같이 확인서 승인 요청이 나타난다. 사용자의 승인이 확인되면 인증기관은 확인서를 전자지갑과 디렉토리에 전송한다.

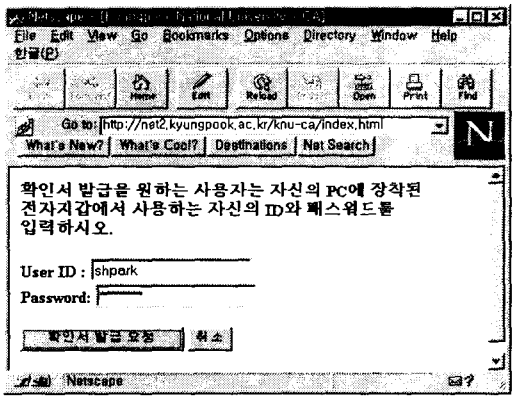


그림 6. 확인서 발급 요청

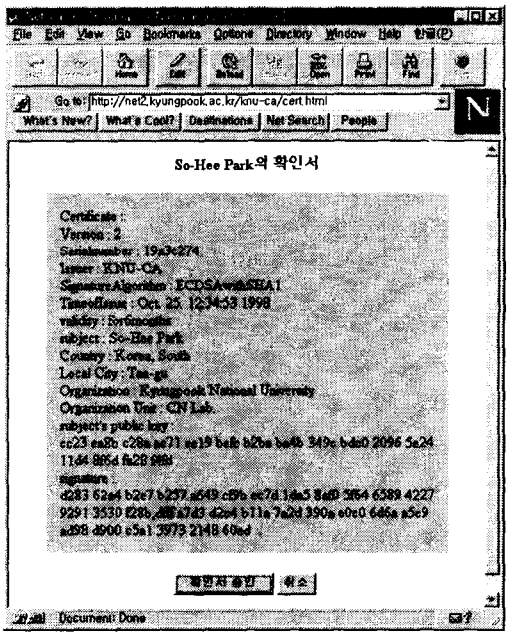


그림 7. 확인서 발급 승인

3. 비교분석

기존의 인증기관은 사용자가 인증기관에 확인서 발급을 요청하면 사용자의 신분을 확인한 후, 자신이 직접 사용자의 공개키/비밀키 쌍을 생성하여 공개키는 확인서를 생성하여 디렉토리에 저장하고 비밀키는 SHTTP[17]나 SSL[29] 등을 이용하여 사용자에게 전자우편을 이용하여 안전하게 전달한다. 그리고, 사용자에게 비밀키를 전달한 후 자신이 가지고 있는 비밀키는 파기하거나, 사용자의 비밀키 분실 시 복구를 대비하여 보관한다. 이러한 경우, 사용자가 비밀키를 분실한 경우 복구는 용이하나, 전자상거래에서 요구되는

비밀키의 안전도를 충분히 제공할 수 없다. 즉, 사용자가 생성한 전자서명이 유일한 사용자의 것임을 믿는 데 한계가 있다.

기존의 전자지갑은 지불과 관련되어 전자화폐를 좀 더 안전하고 효율적으로 사용하기 위해 연구되어 왔다. 그러므로, 신용카드, 직불카드와 같은 여러 가지 지불수단을 수용 가능하게 하는 인증기관의 확인서와 관련된 영역의 연구는 미미한 편이다. 현재 Java Wallet은 신용카드 기반으로 하여 여러 가지 지불수단을 이용할 수 있도록 설계되어 있으나, 아직 확인서와 관련된 부분은 언급되어 있지 않으며 암호시스템의 적용이 이루어지지 않고 있다.

본 논문에서 제안한 인증 프로토콜은 전자상거래에 사용하기에 적합하도록 인증기관과 전자지갑 사이의 연동을 고려하여 기존의 두 기관이 가진 비밀키의 안전한 보관과 신뢰성 문제와 지불 시의 안전성과 효율성 문제를 개선하였다. 즉, 사용자의 전자지갑은 공개키/비밀키 쌍을 직접 생성하고, 비밀키와 확인서를 보관하며 인증기관은 전자지갑이 전달한 공개키에 관한 확인서만을 발급하도록 한다. 암호시스템으로는 타원곡선 암호시스템을 적용하였다. GF($2^{255}$)상의 타원곡선을 이용하였으므로 1024 비트의 소수 p 를 사용하는 RSA보다 높은 안전도를 가진다. 이에 적은 메모리용량과 키 길이를 가지고도 전자상거래 시스템에 이용할 수 있어 효율적이고 앞으로 스마트 카드와 같은 chip card 에도 적용 가능하다.

## VI. 결론

앞으로 전자상거래가 보편화되기 위해서는 여러 가지 요건이 필요하지만, 특히 사용자에게 높은 안전성과 신뢰성이 제공되어야 한다.

본 논문에서는 사용자의 비밀키 안전도를 개선할 수 있는 인증 프로토콜을 제안하기 위해 기존의 인증기관을 기능적 구조별로 분석하여 인증 체계를 연구하였고, 전자지갑의 기본적인 개념과 기존의 전자지갑의 구조적인 측면을 분석하였다. 그리고, 타원곡선 암호시스템의 필요성과 특징 및 장점들을 조사하여 이를 이용한 인증 프로토콜을 제안하였고 구현하였다. 즉, 전자지갑과 연동한 인증기관의 인증 프로토콜은 사용자의 전자지갑이 공개키/비밀키 쌍을 직접 생성하고, 인증기관은 전자지갑이 전달한 공개키에 관한 확인서만을 발급하도록 하여 사용자의 비밀키 안전도를 개선시킬 수 있었다.

향후 더욱 안전하고 신뢰성있는 전자상거래가 실현되기 위해서는 인증기관과 전자지갑 사이의 연동뿐만 아니라, 지불시스템과의 연동이 고려되어야 한다.

## 참고 문헌

[1] B. Schneier, *Applied Cryptography*, 2nd Edition, John Wiely & Sons, 1996.

[2] VeriSign Certification Practice Statement Version 1.2, available at ftp://ftp.verisign.com/reposotory/CPS, 1997.

[3] Java Wallet Architecture White Paper from http://java.sun.com/products/commerce, 1998.

[4] CyberCash Wallet from http://www.cybercash.com/cybercash/wallet, 1998.

[5] A.J. Menezes, P.C. Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.

[6] Federal Public Key Infrastructure Technical Specification Part A: Requirements, NIST, 1996.

[7] Federal Public Key Infrastructure Technical Specification Part B: Technical Security Policy, NIST, 1996.

[8] Federal Public Key Infrastructure Technical Specification Part C: Concept of Operations, NIST, 1996.

[9] Federal Public Key Infrastructure Technical Specification Part D: Interoperability Profile, NIST, 1996.

[10] RFC 1487, X.500 Lightweight Directory Access Protocol, Internet Request for Comments 1487, 1993.

[11] CCITT X.500, The Directory : Overview of concepts, Models and Services, 1992.

[12] NIST, Digital Signature Standard, *FIPS PUB 186*, 1994.

[13] ITU-T Recommendation X.509, Information Technology-Open Systems Interconnection-The Directory : Authentication Framework, 1993.

[14] SET spec. v1.0 from http://www.setco.org/set_specification.html

[15] R. Rivest, A.Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Comm. of ACM, vol.21, no.2, pp.120-126, Feb. 1978.

[16] 김춘길, 전자상거래, KRNET '98 6th Computer Networking Conference 특강자료집, pp.173-200, 1998.

[17] SHTTP from http://www.eit.com/projects/s-http

[18] PEP from http://www.w3.org/TR/WD-http-pep-971121

[19] National Bureau of Standards, "Data Encryption Standard," *NSB FIPS PUB 46*, 1977.

[20] 아이캐시 from http://www.cnkorea.co.kr/

[21] 이니텍페이 from http://www.initech.com/seminar/Payment/index.htm

[22] 메타랜드 전자지갑 from http://www.metaland.com/userguide.html

[23] 넷크레딧 from http://www.cni.co.kr/cni/paper/80803.htm

[24] NIST, "Secure Hash Standards," *FIPS PUB 180-1*, 1995.

[25] A. M. Odlyzko, "The Future of Integer Factorization," *CryptoBytes*, vol.1, no.2, pp.5-12, 1995.

[26] N. Koblitz, "Elliptic Curve Cryptosystems," *Mathematics of Computation*, Vol. 48, No. 177, pp.203-209, 1987.

[27] V. Miller, "Uses of Elliptic Curve in Cryptography," *Advances in Cryptology-Auscrypt '90*, Vol.453, pp.2-13, 1990.

[28] A. J. Menezes, *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers, 1993.

[29] SSL from http://home.netscape.com/newsref/std/SSL.html