다양한 운영체제와 인증서 저장매체를 지원하는 Easy PKI 클라이언트 S/W 설계

박종욱°, 최태규, 이재일
한국정보보호진흥원(KISA) 전자거래보호단

Design of the Easy PKI Client S/W Supporting the Multiple OS Platforms and Certificate Devices

Jongwook Park°, Taekyu Choi, Jaeil Lee
KISA(Korea Information Security Agency)

요 약

본 논문은 인증기관, PKI솔루션 업체, 전자거래기관 위주로 제공되어 온 PKI서비스를 안전성, 편의성, 표준적합성, 이동성, 경제성측면에서 개선된 PKI 클라이언트 S/W 기능과 유저 인터페이스(UI)를 제시한다. 이는 복잡한 PKI기술을 모르는 초보 이용자라 할지라도 USB키, 스마트카드 등 여러 이동형 저장매체를 다양한 운영체제에서 쉽고 안전하게 사용할 수 있게 한다. 특히 제안하는 자동 인증서 업데이트 기능, CD를 활용한 인증서 배포방안, 인증서 공통저장위치 및 파일명명규칙은 하나의 인증서로 다양한 PKI서비스를 이용할 수 있어 Easy PKI화가 가능하리라 기대된다.

1. 서론

공개키기반구조(PKI, Public Key Infrastructure)의 도입은 인터넷을 이용한 전자거래시 이용자 신원확인의 문제를 해결해 주는 역할을 수행한다. 따라서 세계 각국, 조직 등은 앞다투어 PKI서비스를 도입하거나 제공중에 있다. 그러나 PKI개념에 대한 일반인의 이해부족과 서비스 제공자 위주로 구축되어 온 서비스의 복잡함으로 인해 PKI서비스가 활성화되지 못하고 있다. 최근 이러한 문제를 해결하기 위한 방도로 인증서 이용자 환경 개선에 대한 관심이 고조되고 있다[1,2]. 일례로 한국정보보호진흥원(KISA)이 최근 실시한 '전자서명 대국민 인식 및 만족도 조사 결과'에 따르면 전자서명 이용자들은 해킹 등에 의한 외부로의 키 및 공인인증서 유출방지(61.7%), 키 및 인증서 저장방법 UI의 간소화(20.9%), 다양한 인증서 저장매체 지원(16.2%) 등 PKI 클라이언트 S/W 환경개선의 필요성을 공감하고 있다. 이러한 문제발생의 구체적인 원인 중 하나는 다양한 클라이언트 S/W가 동일한 기능에 대해 서로 다른 UI를 적용하고 있는 데에서 찾을 수 있다. 또한 스마트카드, USB키 등 새롭게 등장하는 인증서 저장매체를 발빠르게 수용하지 못하는 점과 차츰 많은 사용자를 확보해 가고 있는 리눅스와 같은 공개 운영체제에서 동작하는 클라이언트 S/W의 부재를 간과할 수 없다.

본 논문에서는 열거한 문제를 해결하는 방안으로 다양한 운영체제 및 이동형 인증서 저장매체를 안전하고 편리하게 사용할 수 있는 Easy PKI 클라이언트 S/W를 설계하고자 한다. 설계원칙은 편의성, 안전성, 표준적합성, 이동성, 경제성이다. 편의성은 누구나 쉽게 사용할 수 있는 직관적인 인터페이스를 제공해야 함을 의미한다. 안전성은 이동형 저장매체에 대해 강조되는 사항으로 매체의 분실, 도난 등으로 인한 개인키 노출과 불법복제시 이에 대한 보호 메커니즘의 보안강도를 제공해야 함을 뜻한다. 표준적합성은 서로 다른 인증도메인간 PKI 상호연동을 위해 필요하며 이동성은 무선 인터넷 환경과 향후 도래할 유비쿼터스 환경에서 최우선으로 고려되어야 할 원칙이다. 마지막으로 이용자의 경제적 부담을 최소로 하기 위해 최소의 투자로 최대의 효과를 거둘 수 있는 경제성 원칙이 적용된다. 본 논문에서는 이러한 설계 원칙으로 인증서 이용자가 클라이언트 S/W를 사용하면서 어려움을 겪는 루트 인증기관 공개키 획득방안, 인증서 저장매체 지원, 인증서 및 개인키 관리방식, 다수 인증서 관리에 대해 고찰하고자 한다.

법으로 영상 데이터에 대하여 워터마크를 집어넣는 방법을 제안한다. 그리고, 4장에서는 제안한 워터마킹 기법에 대하여 시뮬레이션하여 그 유효성을 보인다. 마지막으로 5장의 결론에서는 향후의 연구과제를 제시한다.

2. Easy PKI 클라이언트 S/W 설계

가. 루트인증기관 공개키 자동 업데이트 기능

인증서를 검증하는 신뢰당사자(relying party)는 송신자 인증서를 포함하는 인증경로상의 정점에 있는 루트인증기관 공개키를 안전하게 획득해야 한다. 한편 베리사인(VeriSign), 엔트러스트(Entrust), 써트(Thawte) 등 세계 주요 루트인증기관 인증서는 MS 익스플로러, 넷스케이프와 같은 웹브라우저 프로그램에 기본으로 탑재되어 있다. 즉 상기 인증기관 공개키는 인증기관 프로그램(Certificate Authority Program) 등을 통해 웹 브라우저에 미리 탑재된 후 웹 브라우저와 동시에 설치되어 PKI 개념에 익숙하지 않은 일반 이용자들 대신 자동으로 전자거래 인증서비스에 필요한 신뢰관계를 형성시켜 준다[3]. 그러나 우리나라와 같이 MS 익스플로러나 넷스케이프 브라우저에 미리 탑재되지 않은 제3의(third-party) 루트인증기관 인증서를 신뢰해야 하는 경우 이용자들은 불편을 감수하고 일일이 해당 루트 공개키의 신뢰여부를 결정해야 한다. 또한 이미 신뢰하는 인증서가 만료되거나 갱신되어 신 인증서를 받는 경우에도 마찬가지이다[4]. 이는 PKI 안전성 확보와 이용확산을 위해 지양해야 할 요소로 이용자의 편의성을 충분히 고려한 새로운 루트 공개키의 안전한 배포 방법이 필요하다.

본 논문은 보다 편리하고 안전하게 루트 공개키를 새로 추가하거나 이미 인증서 저장소에 저장되어 있는 루트 공개키들의 정보를 최신으로 유지할 수 있도록 MS 윈도우 운영체제의 '윈도우 업데이트'와 유사한 '자동 인증서 업데이트' 개념을 제안한다. 인증기관은 우선 누구나 공개키 정보를 확인할 수 있도록 관련 정보(인증서, 인증서 해쉬값, 업데이트 목록 등)를 인증기관 웹서버와 디렉토리 서버에 공고한다. 이 때 웹서버와 디렉토리 서버인증을 위해 암호화 통신(SSL, TLS, WTLS)의 사용을 선택적으로 고려할 수 있다. 이용자는 자동 인증서 업데이트 기능을 활용하여 자신이 속한 인증도메인의 루트 공개키를 신뢰하는 절차를 거친다. 유의할 점으로 신뢰절차시 업계에서 대부분 적용하고 있는 클라이언트 S/W에서 보여주는 인증서 해쉬값과 등록기관에서 배부한 인증서 발급요청 확인서에 인쇄된 인증서 해쉬값의 일치여부를 육안으로 확인하는 소극적인 방법을 사용하지 않는다는 점이다. 이는 공격자가 루트 공개키에 대한 스푸핑 공격을 시도할 경우를 대비하기 위함이다[5]. 대신 안전하게 획득한 루트 인증서 해쉬값을 이용자가 직접 입력하여 온라인으로 전달받은 루트 공개키를 신뢰하는 방법을 이용한다. 이러한 방법은 안전성을 강화하는 대신 이용자 편의성을 저해할 수 있으나 다음에 기술하는 인증서 해쉬값의 변환 처리 후 입력하는 방법으로 이용자의 편의성을 제고할 수 있다. 즉 SHA1 해쉬 알고리즘을 적용한 160비트(20바이트)의 인증서 해쉬값을 16진수로 변환한 후 4글자(2바이트)씩 공백(space)을 두어 입력하게 한다면 입력시 발생하는 불편함을 줄일 수 있다. 더욱이 이용자의 의한 인증서 해쉬값의 직접 입력방법은 일반적으로 클라이언트 S/W 초기 설치시 한번만 수행하면 되므로 이용자의 불편을 크게 초래하지는 않을 것이다.

자신이 속한 인증도메인의 루트 공개키를 신뢰하는 이용자는 언제든지 자동 인증서 업데이트 기능을 이용하여 다른 루트 공개키의 유효성을 확인할 수 있게 된다. 특히 신뢰해야 하는 루트 공개키가 다수이거나 자주 갱신되는 경우 이용자는 인증서 업데이트 기능 버튼을 단 한번 클릭으로써 편리하게 루트 공개키 정보를 관리할 수 있게 된다.

나. 인증서 및 개인키 저장매체 지원

컴퓨터 하드웨어 기술이 발전함에 따라 이용자는 기호에 따라 하드디스크, 플로피디스크, CD, USB키, 스마트카드 등 다양한 매체를 PKI서비스에 활용할 수 있게 되었다. 그러나 아직까지 클라이언트 S/W별로 지원하는 저장매체의 종류가 다르고 USB키, 스마트카드 등 일부 매체에 적용되는 표준이 달라 이용자가 실질적으로 사용하기에는 상당한 어려움이 존재한다. 여기서는 안전성, 편의성, 이동성, 경제성, 친밀도 등 여러 기준에서 인증서 저장매체의 장단점을 살펴봄으로써 현재 PKI서비스에 가장 적합한 저장매체가 CD임을 보인다. 제시된 기준을 간략히 살펴보면 다음과 같다. 안전성은 인증서 및 개인키를 보호하기 위한 매체의 물리적인 특성과 소프트웨어적 암호기술의 존재여부를 판단한다. 편의성은 컴퓨터에서 해당 매체를 부착하기 위해 드라이버 설치, 제품 사용매뉴얼 숙독여부 등 이용자가 해당 매체를 통해 PKI서비스를 받

는데 부수적인 제약조건들의 양을 가늠한다. 예를 들어 스마트카드는 스마트카드 리더기가 필요하고 별도의 관리 프로그램을 사용해야 함으로 이용자가 편리하게 사용하기에는 선결해야 문제들이 적지 않다. 반면 이동성은 매체의 크기, 추가 장비 소지필요여부 등 그야말로 이용자가 들고 다니기에 편한 정도를 판단한다. 경제성은 초기구입비용, 유지보수비용, 매체 불량률 등의 파라미터를 통해 이용자의 경제적 부담을 나타낸다. 친밀도는 편의성과 연관이 있는 기준으로 매체 시장의 성숙화 정도, 시장 판매비율 등을 통해 이용자들이 매체 사용시 느끼는 부담감과 반비례한다고 볼 수 있다. 전술한 5개 기준을 종합적으로 고려하여 현재 PKI 서비스에 적용 가능한 인증서 저장매체의 장단점을 통해 현재 및 미래의 사용비중을 추론하면 [표 1]과 같다.

[표 1] 인증서 저장매체별 특징 비교

	하드 디스크	플로피 디스켓	스마트 카드	USB키	CD
안전성	△	×	◎	◎	△
편의성	◎	◎	×	△	○
이동성	×	○	○	◎	○
경제성	×	◎	×	×	○
친밀도	○	◎	×	△	○
현재 사용률	◎	○	×	×	×
향후 사용률	○	×	○	◎	△

기호 : ◎매우 높음 ○높음 △보통 × 낮음

하드디스크는 현재 이용자 PC에 기본적으로 장착되어 있어 별다른 어려움이 가장 많이 이용되고 있는 저장매체이다. 그러나 조만간 도래할 유비쿼터스 환경에서는 이동성, 경제성이 강조됨에 따라 USB키, 스마트카드와 같은 이동형 저장매체가 많이 이용되리라 판단된다. 반면 플로피 디스켓은 편의성, 이동성, 경제성, 친밀도에서 강점이 있다. 그러나 개인키 보호에 있어 하드디스크, CD와 마찬가지로 PKCS#8형태의 소프트웨어 레벨의 암호기법이 적용될 수 있으나 스마트카드와 같이 물리적인 레벨의 보호 메커니즘의 적용은 어렵다. 즉 플로피 디스켓은 개인키에 대한 임의 접근이 가능하므로 분실·도난의 경우 개인키의 노출가능성이 높다. 주지할 점은 플로피 디스켓은 하드디스크와 동일하게 개인키의 접근이 용이하나, 이동성이란 특성 때문에 분실의 위험은 더 높다는 점이다. 또한 플로피 디스켓은 비교적 높은 물리적 매체 불량율·손상률로 안전성측면에서는 미흡한 매체이며 인텔과 마이크로소프트가 2002년 후반부터 플로피 디스켓을 지원하지 않음에 따라 점차 사용비중이 급감하고 있다.

스마트카드는 USB키와 동일하게 안전성 측면에서 사용이 가장 권고되는 매체이다. 즉 개인키 생성과 전자서명 생성·검증이 카드 내부에서만 이루어져 개인키 등 중요 정보에 대한 유출의 위험이 없고 물리적인 불법 접근이 탐지되면 저장매체의 정보가 파괴되는 특징을 갖는다. 그러나 아직까지 일반인이 사용하기에는 불편하고 플로피 디스켓, CD에 비해 초기구입비가 높아 경제적 부담이 있다. 특히 컴퓨터와 부착되는 스마트리더기의 연결 인터페이스는 MS사의 PC/SC, ISO의 직렬(Serial)방식 등 다양한 표준의 존재로 스마트카드를 제대로 인식하기 어려워 일반인들이 CD처럼 사용하기 어렵다.

USB키는 열쇠정도의 크기와 무게로 휴대가 용이하여 최근 플로피디스켓, CD등을 대체하면서 이용률이 높아지는 추세이다. 또한 암호화 표준 상품으로 MS 윈도우, 리눅스, 매킨토시 등 모든 운영체제 플랫폼에서 동작하는데 기술적으로 문제가 없다. 반면 USB키은 USB 타입으로 컴퓨터에 쉽게 부착되나, 별도의 장치 구동 프로그램을 설치해야 이용 가능하므로 초기 설치시 다소의 불편함이 발생할 수 있으나 불편함의 정도는 PC에 랜카드를 설치하는 경우와 비슷하므로 그리 염려할 사항은 아니다. 또한 플로피디스켓, CD매체에 비해 판매가격이 비싸 일반 이용자들에게 경제적 부담을 줄 수 있으나 매체 불량률·손상률이 0%에 가까워 타 매체에 비해 추가적인 비용은 발생하지 않는다.

CD는 모든 기준에서 우수한 특성을 갖는 매체이다. 만일 직경 12cm의 일반CD가 아닌 직경 8cm의 미니 CD를 사용한다면 이동성이 더욱 보장될 것이다. 그리고 CD-R이 아닌 CD-RW를 이용할 경우 매체의 재활용률이 높아질 것이다. 현재 PKI는 많은 장점에도 불구하고 복잡한 정책과 기술로 인해 일반인들이 사용하기 어렵다. 일례로 등록기관에서 오프라인 신원확인후 일반인이 복잡한 인증서 발급절차로 실제 인증서를 수령하는 비율은 미비한 것으로 파악되고 있다. 만일 이용자가 등록기관이 제공하는 PC, CD레코더 등의 물리적 설비를 활용하여 신원확인과 동시에 발급된 인증서를 CD에 저장하여 수령한다면 인증서 발급시 발생하는 인증서 이용자의 불편을 상당부분 해소시켜 스마트카드, USB키 저장매체가 널리 사용하

기 이전의 단계를 잠정적으로 대체할 수 있으리라 판단된다. 또한 CD가 인증서 저장매체로 활성화될 경우, PC포맷·인증서백업의 불편함 등으로 인해 발생하는 인증서 갱신횟수를 상당 부분 줄일 수 있으므로 인증기관의 업무 부담이 경감되는 긍정적인 효과를 볼 수 있으므로 여러 인증서 저장매체 중 CD를 적극 활용할 것을 제안한다.

다. 인증서 및 개인키 관리 방안

일반적으로 웹 브라우저나 MS사의 아웃룩(Outlook)같은 PKI 응용 프로그램은 이용자의 인증서와 개인키를 저장하는 위치 및 방식을 독자적으로 정의하고 있다. 반면 하나의 인증기관으로부터 인증서를 발급받은 이용자는 각각의 프로그램별로 별도의 인증서를 사용하기보다는 기존에 있는 인증서를 활용하여 즉 하나의 인증서로 여러 PKI 응용 프로그램에서 사용하길 원한다. RSA사의 PKCS#12 형식은 이러한 인증서 및 개인키를 교환하기 위한 방식을 정의하고 있다. 그러나 이용자들이 체감하는 불편함은 여전하다. 본 절에서는 이용자 편의성 제고 측면에서 하나의 인증서로 PKI기반의 여러 응용 프로그램을 이용할 수 있도록 인증서 및 개인키 파일의 공통 저장위치, 저장형식, 파일명명규칙을 제안한다.

1) 저장매체별 공통 저장위치

앞에서 고찰한 인증서 저장매체별로 MS 윈도우즈 운영체제와 비 MS 윈도우즈 운영체제 환경에서의 공통 저장위치를 정의하면 [표 2], [표 3]과 같다.

[표 2] MS 윈도우즈 환경에서의 저장위치

하드 디스크	디스크드라이브 레이블[1]:\Program Files\NPKI\인증기관명	
이동식 디스크	플로피 CD USB	드라이브 레이블:\NPKI\인증기관명
스마트 카드	PKCS#15 준용 또는 별도 메모리맵 구성	
USB키	PKCS#15 준용 또는 별도 메모리맵 구성	

1) MS운영체제가 설치된 하드디스크를 말함

[표 3] 비MS 윈도우즈 환경에서의 저장위치

하드 디스크	리눅스	사용자계정[1]/NPKI/인증기관명
	맥OS	디스크드라이브 레이블[2] :\ Library \Preferencence\NPKI\인증기관명
이동식 디스크	플로피 CD USB	(리눅스) 마운트 디바이스명/NPKI/인증기관명 (맥OS) 드라이브 레이블:\NPKI\인증기관명
스마트 카드	PKCS#15 준용 또는 별도 메모리맵 구성	
USB키	PKCS#15 준용 또는 별도 메모리맵 구성	

1) 이용자가 로그온한 계정

2) 매킨토시 O/S가 설치된 하드디스크

2) 인증서 및 개인키 파일 저장형식

저장되는 인증서 및 개인키 파일형식은 인증서 저장매체에 따라 다르게 정의한다. 이는 업계 표준을 준용하여 제공되는 서비스와의 호환성을 유지하기 위함이다. 즉 하드디스크, 플로피디스켓, CD, 윈도우 또는 리눅스 파일시스템(FAT32, NTFS, EXT2 등)을 지원하는 USB드라이브에 저장되는 인증서는 크기 및 여러 PKI기반 응용 프로그램과의 호환성을 고려해야 한다. 이 경우 PEM에서 사용되는 Base64 인코딩 형식의 1/3크기인 DER형식으로 저장한다. 개인키의 경우 패스워드 기반의 암호화 방식인 RSA사의 PKCS#5로 우선 변환한다. 이 때 지원하는 버전은 타 프로그램과의 호환을 위해 버전 1.5 및 버전 2.0을 모두 지원하고 암호화한 후에는 개인키 저장형식인 PKCS#8을 준용하여 안전한 인증서 저장매체에 저장한다. 인증서 및 개인키는 모두 숨김(hidden) 및 읽기전용(read-only) 파일 속성으로 생성하여 이용자의 실수로 인한 파일 삭제를 막을 수 있다.

반면 저장매체가 PKCS#11을 따르는 스마트카드 또는 암호프로세서를 내장한 USB키일 경우 PKCS#11을 준용하여 인증서는 BER형식으로 저장하고 개인키는 바이너리 형태로 저장한다.

3) 인증서 및 개인키 파일 명명규칙

본 절에서는 인증기관 갱신과 다수의 동일 식별명칭(DN)을 갖는 인증서 존재시 인증경로구축을 용이하게 하기 위해 인증기관의 인증서 및 개인키 파일명을 인증서내 소유자 키식별자(SKI, Subject Key Identifier) 확장필드와 인증서 일련번호를 조합한 'SKI_일련번호.der' 형태로 생성할 것을 제안한다. SKI는 소유자 키 식별자 확장필드 값으로 40자리의

16진수로 나타내며 일련번호는 10진수로 표현된다. 일례로 MS윈도우 운영체제에서의 인증기관 인증서 파일의 절대경로는 C:\Program Files\NPKI\KISA\SKI값(16진수)_일련번호(10진수).der로 표현될 것이다.

한편 개인이 소유한 인증서 및 개인키는 인증기관별 디렉토리의 'USER' 디렉토리 밑에 이용자 식별명칭(DN)으로 하위 디렉토리를 생성한 후 각각 'xxx.der'와 'xxx.key' 형태로 저장한다. xxx 명칭은 전자서명용과 암호키분배용을 구별하기 위해 용도별로 동일한 파일명을 사용한다. 식별의 편리함을 위해 전자서명용은 signCert.der, signPri.key로 명명하고 암호키분배용은 kmCert.der, kmPri.key로 부여한다. 리눅스 운영체제의 foo 계정에 저장되는 개인인증서 파일의 절대경로는 전술한 규칙을 따르면 /home/foo/NPKI/KICA/ .signCert.der와 같이 구성된다.

라. 다수 인증서 표시 방법

인증서 이용자는 전자거래시 공통 인증서 저장위치에 존재하는 여러 개의 인증서 중 자신이 사용하고자 하는 인증서를 쉽게 식별하고 사용하길 원한다. 그러한 요구사항을 만족하기 위해 클라이언트 S/W에서 다수 인증서 선택을 위한 UI를 (그림 1)과 같이 제시한다. 인증서 선택 UI는 이용자가 로그인 한 계정에 있는 하드디스크의 모든 인증서를 자동으로 검색하여 보여주되, 해당 인증기관에서 발행한 인증서에 우선순위를 부여한다. 또한 본 고에서 분류한 인증서 저장매체뿐만 아니라 네트워크 드라이브 등 새로이 등장할 저장매체를 지원할 수 있도록 불러오기 기능을 추가하여 인증서 검색기능을 강화하였다. 검색된 인증서는 인증서 소유자명, 만료일, 인증기관명의 인덱스 정보형태로 간략히 표현되어 식별이 용이하도록 구성하였다. 인증서 소유자명은 인증서내의 식별명칭(DN) 중 CN필드를 사용한다. 참고로 하단의 인증서 보기 버튼은 이용자가 선택한 인증서로부터 최상위인증기관까지의 인증경로를 구축하고 경로 유효성을 검증하여 그 결과를 보여주는 기능이다.

(그림 1) 다수 인증서 선택 UI

3. 결론

본 논문에서는 인증기관, PKI솔루션 업체, 인터넷 전자거래기관 등 PKI 서비스기관의 입장이 아닌 일반 인증서 이용자 입장에서 편의성을 강화한 Easy PKI 클라이언트 S/W의 설계를 제시하였다. 이로써 복잡한 PKI기술을 모르는 이용자라 할지라도 USB키, 스마트카드 등 여러 이동형 저장매체를 MS윈도우, 리눅스를 비롯한 다양한 운영체제에서 쉽고 안전하게 사용할 수 있다. 특히 자동 인증서 업데이트 기능, CD를 이용한 인증서 발급방안, 인증서 공통저장위치 및 파일명명규칙은 이용자가 하나의 인증서로 다양한 PKI서비스를 이용할 수 있는 긍정적인 효과를 기대할 수 있을 것이다.

[참고문헌]

[1] Carl M. Ellison, The nature of a useable PKI, Computer Networks 31, pp 823-830, 1999

[2] Peter Gutmann, Plug-and-Play PKI: A PKI your mother can use, USENIX Security Symposium, pp 45-58, 2003

[3] Netscape, Certificate Authority Program, http://wp.netscape.com/security/caprogram/index.html

[4] VeriSign, Intermediate CA Replacement Instructions, http://www.verisign.com. 2004

[5] Zishuang Ye and Sean Smith, Trusted Paths for Browsers: An Open-Source Solution to Web Spoofing, http://www.cs. dartmouth. edu/~pkilab/demos/spoofing/, February 4, 2002