Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.24, NO.1, Feb. 2014 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2014.24.1.31

인증서와 개인키 유출 방지를 위한 보안키 저장소 Secure Key Store

박 영 진,[1†] 김 선 종,[2] 이 동 훈[1‡]
[1]고려대학교 정보보호대학원, [2]이니텍

The Secure Key Store to prevent leakage accident of a Private Key and a Certificate

Young-Jin Park,[1†] Seon-Jong Kim,[2] Dong-Hoon Lee[1‡]
[1]Gradute School of Information Security, Korea University, [2]INITECH CO., LTD

요 약

국내에서는 공개키 기반구조(PKI, Public Key Infrastructure)를 도입하여, 온라인상에서 안전한 정보 전송과 신원확인을 위해서 인증서 기반의 전자서명 인증체계를 구축하여 서비스를 제공하고 있다. 하지만 인증의 기본이 되는 온라인상의 개인 인감 증명서라고 할 수 있는 인증서는 사용자들이 쉽게 접근하고 복사할 수 있는 위치에 저장되어 있어, PC에 설치된 악성 프로그램이이나 웹 계정 해킹 등과 같은 공격에 의해 유출 될 수 있는 위험이 존재한다. 또한 개인키 패스워드는 키보드보안기능을 무력화 시킨 후 로깅 툴 등에 의해서 노출될 수 있기 때문에 인증서 파일이 유출되는 경우, 금전적인 피해와 불법 인증을 통한 사회적인 범죄가 발생할 수 있는 위험이 존재한다. 본 논문에서는 인증서와 개인키 파일 유출로 인한 피해를 예방하기 위해 해당 키 파일들을 Device에 의존적인 키로 암호화함으로서 안전하게 저장하고, 유출 되더라도 다른 Device에서 사용할 수 없도록 하는 기법을 제안한다.

ABSTRACT

In Korea, the Public Key Infrastructure (PKI) has been introduced. For secure information transmission and identification, the electronic signature authorization system of a certificate-based is built, and then the service provide.The certificate is stored in location what users can easily access and copy. Thus, there is a risk that can be stolen by malware or web account hacking. In addition, private key passwords can be exposed by the logging tool, after keyboard security features are disabled. Each of these security weaknesses is a potential conduit for identity theft, property/asset theft, and theft of the actual certificates.

The present study proposes a method to prevent the private key file access illegally. When a certificate is stored, the private key is encrypted by the dependent element of the device, and it is stored securely. If private key leakage occurs, the retrieved key could not be used on other devices.

Keywords: PKI, Device, Device-DNA, Device-DNA Map

접수일(2013년 9월 2일), 수정일(1차: 2013년 11월 15일, 2차: 2014년 2월 11일), 게재확정일(2014년 2월 11일)

† 주저자, pigd5072@korea.ac.kr
‡ 교신저자, donghlee@korea.ac.kr (Corresponding author)

I. 서 론

전자서명법은 전자서명에 관한 기본적인 사항을 정함으로써 국가사회의 정보화를 촉진하고 국민생활의 편익을 증진함을 목적으로 1999년 2월5일 법률 제5792호로 제정되고 2013년 3월23일 법률 제11690호로 일부 개정되었다. 전자서명법은 인증서를 기반으로 하여 전자서명에 법적 인감과 동일한 효력을 부여함으로써, 온라인상의 전자거래를 활성화시키는 제도적 기반을 마련하였고, 전자서명법에 근거하여 2000년 2월 1호 공인인증기관이 지정된 이후 현재 총 5개의 공인인증기관이 지정되어 공인인증서의 발급/관리 업무를 운용하고 있다. 공개키 기반 구조(PKI, Public Key Infrastructure)기반의 인증서는 공개키 알고리즘과 해쉬 알고리즘 등 신뢰되고 안전한 거래를 위해 전 세계적으로 사용되는 암호 알고리즘에 의해 발급되고 사용이 보호된다. PKI서비스에서 인증서는 전자 금융을 거래하기 위한 기본이 되는 접근 수단으로 인터넷 뱅킹, 증권 거래, 인터넷 쇼핑, 전자세금 계산서 등 신뢰를 필요로 하는 온라인 전자 거래 정보의 무결성을 보장하고, 전자거래 행위의 부인방지 기능을 제공한다. 또한 전자서명법에서 인증서를 이용한 전자서명에 대한 법률적 효력을 규정하고 있기 때문에 사이버 상의 많은 사이트들이 신원확인을 위한 최초 인증 수단(log-in)으로 인증서를 사용하고 있다.

하지만 인증서는 파일로 존재하여 접근이 용이하고, 쉽게 복사할 수 있기 때문에 개인 PC에 저장하여 관리를 소홀히 하거나, 이메일 등과 같은 온라인 저장 공간에 보관할 경우 악성 프로그램이나 웹 계정 해킹과 같은 공격으로 인증서 파일이 유출될 수 있는 위험이 존재한다. 최근 실례를 보면, 2012년 12월 사용자의 PC가 공격자로부터 해킹당해 ISP 인증서가 유출되는 사고가 발생하여 금전적인 피해가 발생하였으며 2013년 현재까지 계속 파밍(pharming) 공격을 위한 악성프로그램 또는 피싱(phishing) 사이트 등을 통해 사용자의 인증서가 유출되는 사고가 계속해서 발생하고 있다. 또한 암호화된 개인키 패스워드 실패에 대한 제한이 없기 때문에 단순하게 패스워드를 설정하는 경우에는 사전공격(dictionary attack) 등에 의해 패스워드가 노출되거나, 키보드 보안기능을 무력화시킨 후, 로깅 툴을 이용해서 쉽게 패스워드가 유출될 수 있다.

인증서 저장을 위한 기술 규격에서는 PC하드 디스크에 저장하는 것을 지양하며, USB메모리, 스마트카드, 보안토큰 등과 같은 안전한 이동식 저장매체를 보급하여, 인증서를 해당 매체에 저장할 것을 권장한다. 하지만 해당 매체들을 이용하기 위해서는 추가적인 비용이 요구되며, 사용자들은 인증서 저장 매체에 대한 정확한 인식이 없어 보안 인식 제고를 필요로 하기 때문에 단 시간 내에 효과를 기대하기 어렵다. 또한 추가적인 매체를 소지해야 함으로 인해 사용자들에게 불편함을 초래할 수 있고, 해당 매체를 유실하게 되는 경우에는 동일한 문제를 발생시킬 수 있다

본 논문에서는 PC하드 디스크에 인증서와 개인키를 저장해서 공격자에 의해 유출되더라도 인증서와 개인키를 안전하게 보호함으로서 피해를 예방할 수 있는 기법을 제안한다. 논문의 2장에서 기존 인증서와 개인키가 저장되고 있는 현황과 특징을 살펴보고, 해당 방식의 취약점을 분석한다. 3장에서는 인증서와 개인키를 기존 저장방식보다 안전하게 저장하고, 유출되더라도 공격자로부터 안전할 수 있는 방안(Secure Key Store)을 제시한다. 4장에서는 제안 방안과 기존 방식 대비 안전성과 해당 기술을 배포하기 위한 표준화 방안을 설명하고, 5장에서는 본 논문에 대한 결론을 내린다.

II. 기존인증서와 개인키 저장방식 및 취약점분석

본 장에서는 기술규격에서 명시하고 권장하는 인증서와 개인키 저장 매체의 종류를 살펴보고, 현재 인증서와 개인키를 저장하기 위해 이용하는 매체 현황과 취약점에 대해서 분석한다.

2.1 기존 인증서와 개인키 저장 방식

일반사용자들이 인증서와 개인키를 이용하기 위해서는 오프라인에서 등록기관(RA, Registration Authority)을 통해 신원확인 후 인증기관(CA, Certificate Authority)으로부터 인증서와 개인키를 발급받는다. 이 기능은 보안 업체에서 제공하는 가입자 소프트웨어를 통해서 수행하며, 발급 받은 후 저장하는 방식은 한국인터넷진흥원(KISA, Korea Internet & Security Agency)에서 제공하는 기술 규격에서 명시하고 있다[5].

규격에서는 인증서와 개인키 저장매체를 아래 표와 같이 제시하고, 하드디스크는 인증체계의 전자 서명키를 상향 조정하는 시점(2012년 01월)부터는 사용을 자제할 것을 권고한다. 또한 인증서와 개인키의 발급,

Table 1. storage media kind

구분	저장매체종류	설명
이동식 디스크	플로피디스크 및 CD-ROM USB 드라이브	파일시스템을 지원하는 휴대용 저장매체
보안토큰	보안토큰	전자서명이 하드웨어 기기내부에서 생성되는 저장매체
저장토큰	IC카드	메모리타입 스마트카드
	USB키	독자적 파일구조를 지원하는 USB드라이브
하드 디스크	하드디스크	PC 본체에 내장된 로컬디스크

갱신, 재발급 시 하드 디스크 이외의 저장매체에 저장할 것을 권고 하며, 하드디스크 내에 저장하고자 하는 경우에는 인증서와 개인키에 대한 접근통제 기능, 비밀번호 오류 횟수 제한 기능, 임의 복사 방지 기능 등 안전한 보안 기능을 함께 제공해야 한다고 제시한다[5].

그리고 개인키를 저장할 때는 PKCS #8 에 정의된 EncryptedPrivateKeyInfo형식으로 암호화하여 저장하며[9], 저장 위치는 저장 매체 및 운영체제 별 환경에 따라서 아래 표와 같이 정의하고 있다[5].

Table 2. certificate and private key storage location

저장매체	운영체제		저장위치
하드 디스크	윈도우	98, ME, XP	(하드디스크 레이블명):\ Program Files\ NPKI\(인증기관식별자)
		비스타 이상	%UserProfile% \AppData\LocalLow \NPKI\(인증기관식별자)
	UNIX/Linux		(사용자계정)/NPKI/(인증기관식별자
	Mac OS X		(사용자계정) Library/Preferences/NPKI/(인증기관식별자)
이동식 디스크	윈도우		(드라이브명): \NPKI\(인증기관식별자)
	UNIX/Linux		(마운트 디렉토리)/NPKI/(인증기관식별자)
	Mac		/Volumes/(디스크명)/NPKI/(인증기관식별자)
IC카드			스마트카드 파일 구성도 및 메모리맵 참조
USB키			추후 결정
보안토큰			[KCAC.TS.HSMU] 준용

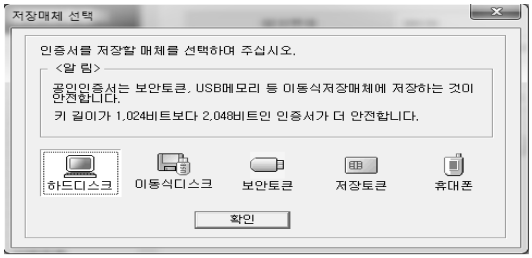

Fig.1. certificate storage medium's selection screen

사용자가 인증서와 개인키의 저장매체로 하드 디스크를 선택하는 경우에는, 하드 디스크가 이동식 디스크, 보안토큰 등에 비해 유출될 가능성이 높아, 안전하지 않은 저장매체임을 공지하는 메시지를 출력해야 한다고 명시한다[5].

2.2 기존 인증서와 개인키 저장 방식의 취약점 분석

2.1에서 살펴본 바와 같이 인증서와 개인키 저장 시 하드 디스크 외에 USB나 보안 토큰 등 다른 매체에 저장할 것을 권고 하지만, 사용자들이 다른 매체를 사용하기 위해서는 추가적인 비용이 발생하게 된다. 또한 저장 매체에 대한 보안 인식 및 매체 사용에 대한 교육이 부족하기 때문에 대부분 하드 디스크에 인증서를 저장한다.

한국인터넷진흥원이 실시한 '2011년 대국민 전자서명 이용실태조사'에 따르면, USB사용도 많지만 여전히 많은 사용자들이 인증서와 개인키의 보관 장소로 자신의 컴퓨터 하드디스크를 선택 한다고 한다[6].

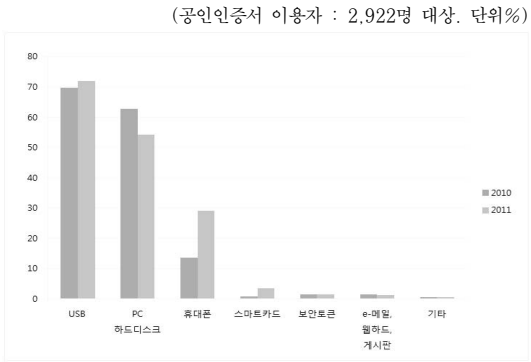

Fig.2. storage media's utilization

인증서와 개인키는 파일(예 : SignCert.der, SignPri.key)형식으로 존재하기 때문에 하드 디스

크의 저장되어 있는 위치에 접근이 쉬우며, 복사해서 사용하기 용이하다. 또한 최근에는 사용자 PC에 설치된 악성 코드로 인한 사용자 인증서와 개인키 파일 및 관련 인증 정보가 유출되는 사고가 빈번하게 발생하고 있다.

아래와 같은 단계로 사용자 PC에 저장되어 있는 인증서와 개인키 파일이 유출될 수 있다.

Step 1. 공격자가 악의적인 목적을 수행하기 위해 웹 사이트를 해킹하여 악성 프로그램을 설치하거나, 일반 사용자의 이메일에 악성 프로그램을 포함하여 메일 등을 보낸다.

Step 2. 사용자가 해킹당한 웹 사이트에 접속하거나, 악성 프로그램이 포함된 메일을 읽는 순간 프로그램이 다운로드 된다.

Step 3. 악성 프로그램이 사용자 PC에 자동으로 설치된다.

Step 4. 악성 프로그램은 사용자 PC에서 잠복하면서 사용자 인증서와 개인키, 그리고 키 로깅 정보를 공격자에게 전송한다.

Step 5. 공격자는 사용자의 인증정보를 모두 취득하여 해당 정보를 이용하여 불법행위가 가능하다.

다음 장에서는 인증서와 개인키를 안전하게 보호함으로서 위와 같은 방식으로 인증서와 개인키가 유출되더라도 다른 Device에서는 사용할 수 없도록 하는 방안을 제안한다.

III. 개인키를 안전하게 저장하는 방안(Secure Key Store)

인증서와 개인키는 파일 형식으로 존재하기 때문에 최초 발급 후, 처음 저장된 Device가 아닌 다른 Device에 복사해서 사용이 가능하다. 정당한 목적에 의해서 복사해서 사용하는 경우에는 사용자들에게 편의성을 제공하지만, 공격자가 악의적인 목적으로 인증서를 복사하는 경우에는 사고를 유발하게 된다. 사고를 예방하기 위해 Device에 의존적인 키를 생성해서 PKCS #5 방식으로 암호화되어 있는 개인키를 이중 암호화함으로서 유출되더라도 다른 Device에서는 사용할 수 없도록 하는 기술을 제안한다.

해당 기술은 Secure Key Store라고 명명하며, Device에 의존적인 키는 동일한 종류의 Device라고 하더라도 각 Device마다 다른 항목으로 조합되기 때문에 Secure Key Store파일이 유출되더라도 쉽게 복호화 할 수 없는 구조이다.

IV. 개인키를 안전하게 저장하는 방안(Secure Key Store)

4.1 Secure Key Store의 개요

4.1.1 용어 정리

(1) Device : 인증서와 개인키 파일을 저장할 수 있는 기기.

(2) Device-DNA : Device에서 수집할 수 있는 요 소를 나타냄. 예를 들면 MAC Address 나 HDD 일 련번호 등 각각을 Device-DNA 라고 함.

(3) Device-DNA Code : Device-DNA값을 식별하기 위한 유일한 값.

(4) Device-DNA Value : Device의 Device-DNA에 해당하는 값. 예를 들면 MAC Address의 값 '00-0D-F0-82-D5-47, 00-0A-A9-60-B5-00'

(5) Device-DNA Map : Device에서 수집가능한 모든 요소들 중에서 랜덤하게 추출한 Device-DNA Code 집합.

(6) 식별자 : Device-DNA Value의 변경여부를 체크하기 위해 처음 수집할 때 해쉬 등을 이용하여 식별하기 위한 값을 생성함.

4.1.2 Secure Key Store 개요

(1) Device 내 수집 가능한 DNA 요소 검색 : Device에서 추출 가능한 정보를 수집하며, 추출할 수 있는 요소를 각각 Device DNA이라고 한다. (예: MAC Address, HDD일련번

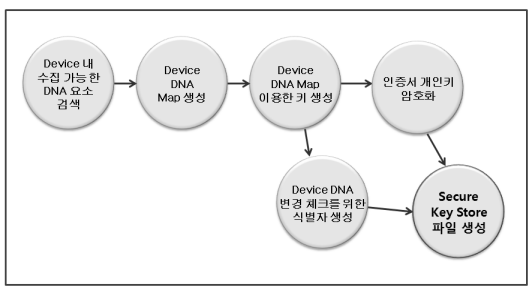

Fig.3. Secure Key Store generation principle

호, HDD총용량 등)

(2) Device DNA Map 생성 : (1)의 정보에서 랜덤하게 요소를 추출해서 Map을 생성한다.

(3) Device DNA Map 이용한 키 생성 : Device DNA Code에 해당하는 Value를 수집하고, 해당 값들을 이용하여 PKCS #5 PBKDF2 방식으로 키를 유도한다[9].

(4) Device DNA 변경 체크를 위한 식별자 생성 : Device DNA는 변경될 수 있기 때문에 Secure Key Store를 암호화 한 키 변경 여부를 확인하기 위해 식별자(예: 일방향 해쉬값)를 생성한다.

(5) 인증서 개인키 암호화 : (3)에서 생성한 키를 이용해서 인증서와 개인키를 암호화한다.

(6) Secure Key Store파일 생성 : 암호화한 값을 파일로 저장한다.

Secure Key Store가 생성되는 단계를 간략하게 설명했으며, 다음은 각 단계에서 수행하는 기능을 상세하게 설명한다.

4.2 Device에 의존적인 키 생성

Device-DNA는 Device를 식별할 수 있는 정보를 수집한 각각의 요소를 의미하며, 각 Device-DNA에 해당하는 Value를 추출하여 암호화를 위한 키를 생성한다.

Device-DNA Value를 이용하여 키를 생성하는 단계는 아래와 같다.

Step 1. Device에서 수집 가능한 Device-DNA를 구성하고, 그 값을 식별할 수 있는 Device-DNA Code와 매핑된다.

Step 2. Device-DNA Code를 랜덤하게 추출하여 Device-DNA Map을 생성한다.

Step 3. Device-DNA Map의 Code에 해당하는 Device-DNA Value를 수집하여, 키를 생성한다.

4.2.1 Device-DNA Map 생성

인증서가 저장된 Device에서 안정적으로 추출할 수 있는 정보를 수집하여 유일하게 식별할 수 있는 Device-DNA Code를 이용하여 Device-DNA-MAP를 구성한다.

Table 3. device informations

Code	Device-DNA
39A2	HDD(C드라이브) 볼륨 일련 번호
A3D7	HDD(C드라이브) 저장소 총 용량
3B19	HDD(C드라이브) 파일 시스템
72AB	HDD 장치 드라이버 이름
CD3A	MAC Address 모음
24C9	Network Adaptor 이름 모음
1D4A	Processor 이름
8A12	Window Version
5193	Monitor 이름
4CAB	Computer이름
6B18	Momory 정보

Table 4. Device-DNA Map example

Device-DNA Code
39A2
72AB
CD3A
1D4A
8A12

Device-DNA Map은 Device-DNA Code를 랜덤하게 추출하여 구성하고, Map의 구성은 주기적(예: 한달, 일년 등)으로 변경되도록 해야 한다

Device-DNA Map을 구성하고 있는 Code에 해당하는 Device-DNA Value를 추출하기 위해 사용가능한 API는 아래와 같다.

Table 5. API for PC device informations

API	설명
GetVolumeInformation	지정된 드라이브에 대한 볼륨이름, 볼륨 시리얼 넘버, 파일시스템 이름 출력 BOOL WINAPI GetVolumeInformation(_In_opt_ LPCTSTR lpRootPathName, _Out_opt_ LPTSTR lpVolumeNameBuffer, _In_ DWORD nVolumeNameSize, _Out_opt_ LPDWORD lpVolumeSerialNumber, _Out_opt_ LPDWORD lpMaximumComponentLength, _Out_opt_ LPDWORD lpFileSystemFlags, _Out_opt_ LPTSTR

API	설명
	lpFileSystemNameBuffer, _In_ DWORD nFileSystemNameSize);
GetDiskFreeSpaceEx	하드디스크의 총 용량 출력 BOOL WINAPI GetDiskFreeSpaceEx(_In_opt_ LPCTSTR lpDirectoryName, _Out_opt_ PULARGE_INTEGER lpFreeBytesAvailable, _Out_opt_ PULARGE_INTEGER lpTotalNumberOfBytes, _Out_opt_ PULARGE_INTEGER lpTotalNumberOfFreeBytes);
GetAdaptersAddress	컴퓨터 Mac주소 출력 ULONG WINAPI GetAdaptersAddresses(_In_ ULONG Family, _In_ ULONG Flags, _In_ PVOID Reserved, _Inout_ PIP_ADAPTER_ADDRESSES AdapterAddresses, _Inout_ PULONG SizePointer);
GetMonitorInfo	모니터 이름 출력 BOOL GetMonitorInfo(_In_ HMONITOR hMonitor, _Out_ LPMONITORINFO lpmi);
GetVersionEx	Windows 버전 출력 BOOL WINAPI GetVersionEx(_Inout_ LPOSVERSIONINFO lpVersionInfo);
GetSystemInfo	CPU 프로세서 유형 출력 void WINAPI GetSystemInfo(_Out_ LPSYSTEM_INFO lpSystemInfo);
GetComputerName	컴퓨터 이름 출력 BOOL WINAPI GetComputerName(_Out_ LPTSTR lpBuffer, _Inout_ LPDWORD lpnSize);
GlobalMemoryStatusEx	컴퓨터 가상 메모리, 실제 메모리(물리 메모리)정보 출력 BOOL WINAPI GlobalMemoryStatusEx(_Inout_ LPMEMORYSTATUSEX lpBuffer);

4.2.2 Device-DNA Map을 이용한 키 생성

Device-DNA Map의 Device-DNA Value를 이용해서 PKCS #5 PBKDF2 방식으로 키를 유도한다.[10]

- salt : random value
- iteration : 1000 (최소 1000이상)
- hmac algorithm : HMACwithSHA256(len : 32)
- password : Device-DNA Values

DK = PBKDF2 (password, salt, algorithm, iteration count, dklen)

아래 그림과 같이 PKCS #5 PBKDF2 방식으로 키 유도 시 32 Byte로 키를 생성하고, 키를 16 Byte 씩 잘라서 Key와 Initial Vector 로 사용한다.

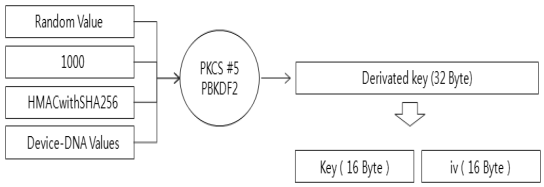

Fig.4. key generation using Device-DNA values

4.2.3 주기적인 키 변경

키에 대한 보안을 강화하기 위해서 키를 주기적(예 : 한달, 일년 등)으로 변경한다. 인증서와 개인키를 Secure key Store에 저장한 시점을 기준으로 주기를 체크해서, 설정한 주기가 되면 복호화 후 새로운 키를 생성해서 다시 암호화 후 저장한다.

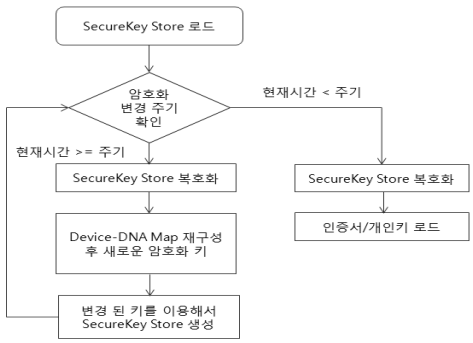

Fig.5. periodic key change

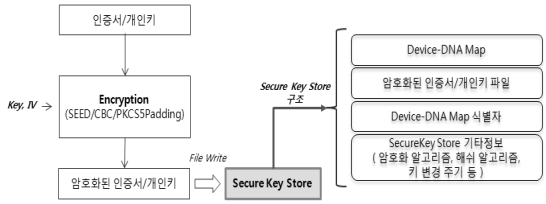

Fig.6. Secure Key Store generation

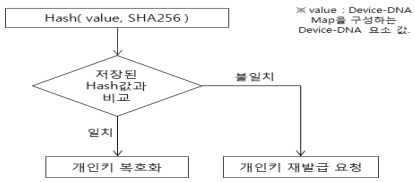

Fig.7. Device-DNA Value change check

4.3 Secure Key Store 생성 및 관리

Secure Key Store는 인증서와 개인키를 저장할 때 Device에 의존적인 정보를 이용해서 이중 암호화 되기 때문에 다른 Device에서 로드 할 수 없다.

4.3.1 Secure Key Store 생성

Device-DNA Map을 통해 유도된 키를 이용해서 사용자 인증서와 개인키를 암호화 후 Secure Key Store를 생성한다.

- Key / Initial Vector : Device-DNA 통해 유도된 키
- 알고리즘 : SEED/CBC/PKCS5Padding

4.3.2 Device-DNA Map 식별자를 이용한 변경 체크

암호화된 인증서와 개인키를 다른 Device에서 복호화를 시도하거나, 동일 Device라고 하더라도 HDD나 CPU등 부품이 변경된 경우에 Device DNA Value의 값이 변경된다. 이런 경우에는 기존 암호화된 인증서와 개인키를 복호화 할 수 없게 되며, 보안을 위해서 사용자에게 다시 발급 받도록 알림을 주어야 한다. Device-DNA Value의 변경 여부를 확인하기 위해 최초 키 생성 시 해당 Value를 해쉬데이터로 저장하고, 키를 로드 할 때마다 해당 값을 비교해서 일치하지 않는 경우에는 Device-DNA Value정보가 변경된 것으로 판단한다.

4.4 다수 Device에서 Secure Key Store 사용하는 방안

하나의 Device를 사용하는 것보다 보안성은 떨어지지만, 사용자의 편의성을 위해 다른 PC나 모바일 등 다수의 Device에서 Secure Key Store를 사용할 수 있는 방안을 제시한다.

첫째, USB나 스마트 카드 등과 같은 이동식 매체를 이용하여 Secure Key Store를 구성하는 방안이다. PC에서 Device-Map을 추출해서 키를 생성하는 것과 동일한 방식으로 이동식 매체에 적용해서 Secure Key Store를 생성하고, 인증이 필요한 경우에는 이동식 매체를 이용한다. 단, 이동식 매체에서 추출할 수 있는 고유 정보는 PC에서 추출하는 정보에 비해 적어 키를 생성하는데 보안성이 떨어질 수 있으며, 매체가 유실 되지 않도록 보관에 특히 유의해야 한다.

둘째, 중계 서버를 이용해서 다른 PC나 모바일 Device에서 Secure Key Store를 이용하는 방안이다. 인증서를 발급 받은 Device에서 인증서 내보내기를 시도하면 Secure Key Store가 복호화 후 중계 서버로 전달되고, 중계 서버에서는 기기를 인증하기 위한 인증코드를 생성하며 해당 값으로 인증서와 개인키(PKCS #8 EncryptedPrivateKeyInfo)를 암호화 후 메모리에 저장한다. 인증서 가져오기를 수행할 Device에서는 인증서와 개인키를 받아 Secure Key Store 구조로 저장한다.

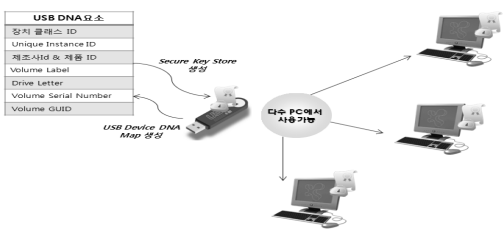

Fig.8. example for secure key store use method for multi device by using removable storage device(USB)

V. 제안한 기법 분석

5.1 제안된 기법의 안전성 분석

개인키는 PKCS #5 방식을 따라 암호화 되어 있

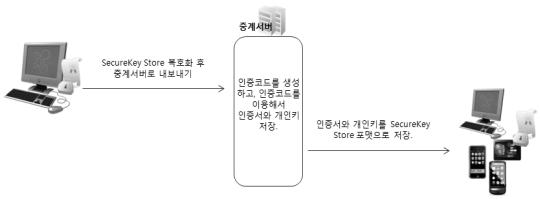

Fig. 9. example for secure key store use method for multi device by using relay server

고, 사용할 때마다 사용자는 패스워드를 입력하고, 패스워드가 일치하는 경우에만 사용이 가능하다.

하지만 사용자 PC에 공격자가 배포한 악성 프로그램이 설치된 경우, 패스워드를 입력하는 순간 키보드 보안 기능이 적용되어 있더라도 해당 기능이 무력화되어, 패스워드가 키 로깅 툴에 의해서 노출 될 수 있고, 키 로깅 내용과 개인키 파일이 공격자에게 유출될 수 있다. 그리고 공격자는 해킹한 정보를 이용해서 자신의 Device에서 타인의 인증서와 개인키를 로드하고 웹 사이트 로그인이 가능해진다.

Fig.10. key logging example

기존 인증서 저장 방식에 이와 같은 취약점이 존재하며 Secure Key Store기술을 제안함으로서, 다음과 같은 안전성을 제공한다.

첫째, 암호화 되어 있는 인증서와 개인키를 Device-DNA를 이용해서 이중 암호화함으로서 기밀성을 강화한다. Device마다 다른 비밀키를 이용하여 이중 암호화함으로서 한번만 하는 암호화 방식보다 강한 안전성을 제공한다.

둘째, Device에 의존적인 키로 인증서와 개인키를 암호화함으로서 유출되더라도 다른 Device에서 로드할 수 없도록 한다. 또한 Device마다 동일한 항목으로 암호화키를 생성하지 않고, 랜덤한 다른 항목을 이용해서 키를 유도하기 때문에 강한 안전성을 제공한다.

셋째, Secure Key Store 파일을 USB와 같은 이동식 매체에 저장하더라도, 지정된PC에서만 인증서와 개인키를 로드 할 수 있기 때문에, USB를 유실하거나 다른 PC에서 로드하려고 시도하더라도 사용할 수 없다.

마지막으로 공격자가 키를 찾아내는데 걸리는 시간보다 짧은 주기로 키를 변경함으로서 Secure Key Store를 암호화 한 키가 유출 되지 못하도록 보호한다. 또한 공격자가 Secure Key Store 프로그램을 역공학(Reversing Engineering)기술을 이용해서 소스 분석 후 Key를 유출할 수 있기 때문에. 이를 방어하기 위해 packing이나 protection기술 등을 이용해서 코드 난독화를 적용한다.

Table 6. Packing/Protection support technology

method	support
Packing	· advanced processing of executable files (EXE, DLL, OCX) · encoding and compression of program code, data, and resources · completely transparent, self-contained operation with long filename support · fast decompression routines deliver better performance than competing products · integrates directly into Windows as a shell extension for ease of use
Protection	· compression of the application · encryption of the application · counteraction to dumping application memory using tools like ProcDump. · application integrity check · API for interaction between application and protection routines · creation and verification of registration keys using public keys encryption algorithms

5.2 제안된 기법을 배포하기 위한 표준화 방안

인증서 저장관련 기술 규격에서는 하드디스크 내에 인증서를 저장하고자 하는 경우 인증서 및 개인키에 대한 접근 통제 기능, 임의 복사 방지 기능 등 안전한 보안 기능을 함께 제공해야 한다고 제시한다[5]. 그리고 개정된 전자서명법 제 16조(공인인증서의 효력의 소멸 등) 부분에서 〔제10조의 규정에 따라 인증업무

를 휴지 또는 폐지하였거나 제12조의 규정에 따라 인증업무가 정지된 공인인증기관의 전자서명생성정보가 분실·훼손 또는 도난·유출되는 등의 경우에는 인증업무의 안전성과 신뢰성 확보를 위하여 해당 공인인증기관이 발급한 모든 공인인증서의 효력을 정지할 수 있다.〕 라고 명시한다[3]. 규격에서는 인증서의 생명주기 관련해서 정책을 정하고 있지만, 인증서 도난, 유출 후에 대한 방안에 대해서만 고지하고 인증서를 안전하게 저장하고 보관할 수 있는 명시적인 방안은 존재하지 않는다.

최근 파밍 공격 등을 목적으로 하는 악성 프로그램이 설치되어 사용자 인증서가 유출되는 사고가 자주 발생하고 있다. 인증서 유출과 관련해서 ActiveX의 취약점, 공인인증서를 대체할 수 있는 방안 등에 대한 논의가 끊이지 않지만, 지금 발생하는 문제들을 당장 해결하기 위한 방안을 도출하기는 쉽지 않다. 또한 한국인터넷진흥원이 실시한 '2011년 대국민 전자서명 이용실태조사'에 따르면 [6], 인증서 사용에 대한 사용자의 인식이 보안을 위해서 사용하는 것이 좋다고 생각하고 있음을 보여주기 때문에 인증서를 사용하되 안전하게 사용할 수 있는 정책을 마련해야 한다.

(공인인증서 이용자 : 2,922. 단위 %)

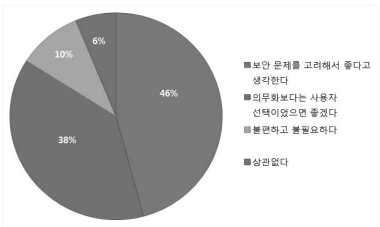

Fig.11. recognition for certificate's use

즉, 인증서를 하드디스크가 아닌 다른 Device 에 저장하더라도 유출될 위험은 항상 존재하기 때문에 인증서가 유출되더라도 안전을 보장할 수 있는 방안이 필요하다.

본 논문에서 제안하는 방안은 현재 사용하는 인증서가 제공하는 기능과 편의성을 유지하면서, 인증서가 유출되더라도 안전할 수 있는 보안 기술이다. 인증서를 하드웨어에 저장하는 것을 지양하기 보다는 제안 기술과 같이 안전하게 개인키와 인증서를 관리할 수 있는 방안을 도입하여, 인증서 저장관련 보안 기능 제공을 권고가 아닌 필수 사항으로 공지하여 모든 가입자 소프트웨어에 Secure Key Store와 같은 보안 장치가 포함되도록 해야 한다. 또한 보안 USB, 보안 토큰, 스마트카드 등과 같은 물리적 매체를 사용하지 않으면서 안전성을 보장할 수 있기 때문에 제안 기술은 추가적인 부담 없이 보급이 가능한 이점이 있기 때문에 해당 기술과 같은 규격에 대한 표준화가 필요하다.

VI. 결 론

본 논문에서는 인증서 저장 방식에 대한 기술 규격을 알아보고, 인증서를 PC하드 디스크에 저장했을 때의 취약점을 분석했다. 그리고 인증서와 개인키가 파일의 형식으로 존재하기 때문에 쉽게 복사 되고, 악성 프로그램 등에 의해 쉽게 유출될 수 있으며, 키보드보안 무력화와 키 로깅 툴에 의해서 개인키 패스워드가 유출될 수 있기 때문에 인증서가 유출되어도 안전할 수 있는 방안을 제시했다. Secure Key Store는 사용자의 개인키를 Device에 의존적인 정보를 이용해서 암호화함으로서, 키 파일들이 유출되어도 다른 Device에서는 사용할 수 없도록 설계되었기 때문에 도난이나 유출됨에 따라 발생 할 수 있는 사고를 예방할 수 있음을 보여주었다. 제안된 기법은 현재 사용되는 인증 표준 체계를 그대로 따르면서, 물리적 매체를 보급하기 위한 시간과 금전적인 추가 부담 없이 단시간 내에 적용이 가능하며 공격자에 의해 파일과 패스워드가 모두 노출되더라도 불법적인 사용을 방지할 수 있고, 이동식 저장매체나 다른 매체를 사용하는 경우에 비해 사용자가 쉽고 편리하게 이용할 수 있는 장점이 있음을 알 수 있다. 현재 인증서를 PC의 하드디스크에 보관함으로서 발생할 수 있는 파일 도난, 파일 유출로 인해 발생하는 신원 도용과 같은 불법 사용의 문제에 대해서 보안 USB, 스마트카드, 보안 토큰 등의 안전한 저장 매체를 보급하고 이러한 매체의 사용을 장려하는 것과 동시에 본 본문에서 제시한 기법을 구현한 소프트웨어를 보급함으로써 온라인 쇼핑몰, 인터넷 뱅킹 등과 같은 온라인 상의 여러 어플리케이션에서 안전하게 인증서를 이용 할 수 있는 환경을 조성하는데 기여를 할 수 있을 것으로 기대한다.

References

[1] Hyung-uk Kim, "A Enhanced Private Key Protection Techniques in the Device Authentication Environments," M.A,Soongsil University, Dec. 2011

〔2〕 Ho-kuen Lee, "public certificate management techniques by random number information for identification," Korea University ,M.A,Dec.2010
〔3〕 National Assembly, "Digital Signature Act(DSN)", Mar, 2013
〔4〕 Korea Internet & Security Agency, "KCAC.TS.CM-Certificate Management in Mobile Device," v1.30, Feb, 2012
〔5〕 Korea Internet & Security Agency, "KCAC.TS.UI-User Interface Specification for the Interoperability between Accredited Certification Authorities," v1.83, Feb. 2012
〔6〕 Korea Internet & Security Agency, "Research on the Actual condition of Electronic Signature System Usage," Dec. 2011
〔7〕 Byung-hoon Kang, Beom-soo Kim and Kyung-kyu Kim, Society for e-Business Studies, "Securing the Private Key in the Digital Certificate Using a Graphic Password," 16(4), pp.1-16, Nob. 2011
〔8〕 Ki-jung Lee , Tae-kyoung Kwon, Seong-woon Hwang and Ki-song Yoon, Jonornal of The Korea Institute of information Security & Cryptology, "A Study on the Secure Storage Device for Protecting Cryptographic Keys in Untrusted DRM Client Systems," 14(2), pp.3-13, Apr.2004
〔9〕 Network Working Group, "Public-Key Cryptography Standards (PKCS) #8 : Private-Key Information Syntax Specification v1.2," RFC 5208, May. 2008
〔10〕 RSA, "PKCS #5 v2.0 : Password-Based Cryptography Standard", Mar.1999

〈저자소개〉

박 영 진 (Young Jin Park) 정회원
2005년 2월: 울산대학교 공과대학 컴퓨터정보통신공학부 학사 졸업
2005년 7월~현재: 이니텍 보안개발본부 과장
2012년 3월~현재: 고려대학교 정보보호대학원 정보보호학과 석사 과정
〈관심분야〉 암호프로토콜, 금융정보보안

김 선 종 (Seon Jong Kim) 정회원
2007년 2월: 울산대학교 공과대학 컴퓨터정보통신공학부 학사 졸업
2007년 9월~2010년 3월: 고려대학교 정보경영공학전문대학원 정보보호 전공 석사 졸업
2004년 6월~현재: 이니텍 미래기술연구소 차장
〈관심분야〉 암호프로토콜, 금융정보보안

이 동 훈 (Dong Hoon Lee) 종신회원
1983년 8월: 고려대학교 경제학사 졸업
1987년 12월: Oklahoma University 전산학과 석사 졸업
1992년 5월: Oklahoma University 전산학과 박사 졸업
1993년 3월~1997년 2월: 고려대학교 전산학과 조교수
1997년 3월~2001년 2월: 고려대학교 전산학과 부교수
2001년 3월~현재: 고려대학교 정보보호대학원 교수
〈관심분야〉 암호프로토콜, 암호이론, USN이론, 키 교환, 익명성 연구, PET 기술