*Journal of The Korea Institute of Information Security & Cryptology* ISSN 1598-3986(Print)
VOL.24, NO.5, Oct. 2014 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2014.24.5.781

# 서명용 개인키 노출 탐지 기법*

박 문 찬,† 이 동 훈‡
고려대학교 정보보호대학원

# A Method for Detection of Private Key Compromise*

Moon-chan Park,† Dong-hoon Lee‡
Graduate School for Information Security, Korea University

## 요 약

PKI(Public Key Infrastructure)는 공개키 암호시스템을 안전하게 사용하고 관리하기 위한 보안 표준방식이다. 인터넷상의 전자 금융거래와 같이 안전하지 않은 채널 상에서 사용자의 전자 서명, 인증, 암호화 등과 같은 보안 기능이 요구되는 환경에서 PKI를 채택하고 있다. 현재 PKI에서 소프트 토큰 기반의 개인키는 표준화된 저장소에 파일 형태로 저장되어 있기 때문에 유출되기 쉬우며, 패스워드 기반 암호화 방식으로 보호되어 있으므로 전수조사 공격에 취약하다.

본 논문에서는 소프트 토큰 기반의 개인키 파일이 유출되더라도 패스워드 전수조사 공격에 확률적으로 안전한 서명용 개인키 노출 탐지기법을 제안한다. 제안 기법은 하나의 올바른 공개키/개인키 쌍과 $n-1$개의 위장(fake) 공개키/개인키 쌍을 사용함으로써 공격자가 패스워드 전수조사 공격에 성공하더라도 올바른 서명 값 생성 또는 인증 성공 확률을 $\frac{1}{n}$로 낮춘다. 이는 공격자가 위장 개인키로 인증을 시도할 경우 이를 탐지하여 해당 인증서를 폐지하고 사용자에게 통지해 주는 기능도 포함한다. 마지막으로 기존 PKI 및 SSL/TLS를 확장하여 제안 기법을 사용할 수 있게 함으로써 추가 인프라 구축비용 없이 소프트 토큰 기반 개인키 저장 방식의 보안 강도를 높일 수 있도록 한다.

## ABSTRACT

A Public Key Infrastructure (PKI) is security standards to manage and use public key cryptosystem. A PKI is used to provide digital signature, authentication, public key encryption functionality on insecure channel, such as E-banking and E-commerce on Internet. A soft-token private key in PKI is leaked easily because it is stored in a file at standardized location. Also it is vulnerable to a brute-force password attack as is protected by password-based encryption.

In this paper, we proposed a new method that detects private key compromise and is probabilistically secure against a brute-force password attack though soft-token private key is leaked. The main idea of the proposed method is to use a genuine signature key pair and $(n-1)$ fake signature key pairs to make an attacker difficult to generate a valid signature with probability $1/n$ even if the attacker found the correct password. The proposed method provides detection and notification functionality when an attacker make an attempt at authentication, and enhances the security of soft-token private key without the additional cost of construction of infrastructure thereby extending the function of the existing PKI and SSL/TLS.

**Keywords:** Public Key Infrastructure, key compromise detection, certificate revocation, revocation notification

접수일(2014년 6월 3일), 수정일(2014년 9월 11일), 게재확정일(2014년 9월 25일)

* 이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(No. 2010-0020726)

† 주저자, rudrnrwlska@naver.com

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

## I. 서 론

공인인증서는 공개키에 소유자 정보를 추가하여 만들어진 일종의 전자거래용 인감증명서이다. 이는 전자금융 거래 시 사용자의 신원확인, 거래 내역 위·변조 방지, 거래사실의 부인 방지 등의 기능을 제공한다. 이러한 보안적 기능으로 사용자는 인터넷 상에서 안전하게 금융 거래를 할 수 있는 환경이 되었다. 시간 및 장소에 관계없이 인터넷을 이용하여 은행업무, 증권업무, 쇼핑, 민원서비스 등 일상생활의 대부분의 업무를 볼 수 있다는 편의성으로 인해 다양한 분야에서 보편적으로 사용되고 있다. 2012년 6월 기준 공인인증서 사용자 수는 2,795만 명으로 집계되고 있다[19].

공인인증서가 일상생활에 널리 쓰이는 만큼 공인인증서와 관련된 보안 문제점도 제기되어 왔다[7]. 그 중 대표적인 것이 개인키 유출 문제이다. 현재의 PKI 구조에서는 인증기관 간 상호연동성을 제공하기 위해 공개키 인증서와 개인키 파일을 표준화된 위치에 저장하도록 하고 있다[17]. 개인키 파일은 표준화 된 위치에 파일형태로 저장되어 있어 개인키 파일 접근과 복제가 용이하다. 최근 들어 언론에 공개되고 있는 공인인증서 유출 사고에서 볼 수 있듯이 개인키 파일은 어렵지 않게 유출될 수 있다. 개인키 파일은 소프트 토큰 방식인 표준화된 패스워드 기반 암호화[13]로 보호되고 있어 유출 자체가 위협이 되는 것은 아니다. 그러나 패스워드 기반 방식은 근본적으로 전수조사 공격에 취약하고, GPU(Graphics Processing Unit)를 PBKDF2[13] 기반 패스워드 크랙(crack)에 활용한 최신 기술[2,6]과 패스워드 크랙 알고리즘[14]에 관한 연구 결과는 사용자 개인키 파일이 유출되었을 경우 패스워드 크랙을 통해 어렵지 않게 사용자의 개인키가 노출(key compromise)될 수 있음을 뜻한다.

최근 이러한 소프트 토큰 방식으로 저장되어 있는 개인키 파일 유출 가능성으로 인해 금융권에서는 하드웨어 기반 토큰 방식인 보안토큰 도입을 추진하고 있지만 추가 비용 문제와 소지의 불편함으로 인해 현재까지는 보급률이 낮은 실정이다. 최근 조사 결과[18]에 따르면 공인인증서 사용자 중 스마트카드 이용률은 3.6%, 보안카드 이용률은 1.5% 이다. 그에 반해 사용자의 절반 이상인 54.3%가 인증서 저장에 PC의 하드디스크를 이용하고 있다.

사용자 개인키 파일이 노출되었을 경우 해당 인증서를 폐지함으로써 이차 사고를 방지할 수 있다. 그러나 개인키는 파일형태로 저장되어 있어 복제로 인한 유출 사고가 발생하였을 경우 사용자가 이를 인지하기란 사실상 불가능하다. 따라서 현 PKI 구조에서는 인증서 노출이 의심되는 경우 사용자가 폐지 신청을 할 수 있게 되어 있으나 소프트 토큰 방식에서는 이런 폐지 기능이 무용지물이다. 즉 현재의 소프트 토큰 방식에서는 개인키 파일이 유출되어도 이를 자동으로 탐지할 수 있는 기술은 현재까지 조사되지 않고 있다.

본 논문에서는 PKI에서 소프트 토큰 기반의 개인키 파일이 유출되어도 패스워드 공격에 안전한 서명용 개인키 노출 탐지기법을 제안한다. 제안 기법의 주요 아이디어는 올바른 공개키/개인키 쌍과 $n-1$개의 위장 공개키/개인키 쌍을 생성하여 사용자 디바이스 저장 공간에 저장하게 하는 방식이다. 유출된 개인키 파일에 대해서 공격자가 패스워드 크랙에 성공하더라도 올바른 개인키 식별을 어렵게 구성함으로써 공격자가 올바른 서명 값을 생성 또는 개인키를 이용한 인증에 성공할 확률을 $\frac{1}{n}$로 낮춘다. 또한 공격자가 위장 개인키로 인증을 시도할 경우 이를 탐지하여 해당 인증서를 폐지시킬 수 있는 PKI 구조에서의 새로운 기법을 제안한다. 이는 공격자가 위장키로 인증을 시도할 경우 $1-\frac{1}{n}$확률로 이를 탐지하여 인증서를 폐지하고 이를 사용자에게 통지해 줄 수 있는 기법이다. 마지막으로 기존 PKI 구조 및 SSL(Secure Socket Layer) / TLS(Transport Layer Security), OCSP(Online Certificate Status Protocol)에 제안 기법을 적용할 수 있게 확장함으로써 추가 인프라 구축비용 없이 소프트 토큰 기반 개인키 저장 방식의 보안 강도를 높일 수 있도록 한다.

논문의 구성은 다음과 같다. 2장 배경지식에서는 공개키 기반구조의 전반적인 내용을 서술하고, 현재 PKI 구조의 문제점과 개인키 유출 위협에 대해서 서술한다. 3장 개인키 유출 탐지 기법에서는 제안 기법 설계를 위한 보안 요구사항을 정리하고 이에 맞춰 기존 PKI에 확장하여 적용할 수 있는 방법에 대해서 제안한다. 4장 안전성 분석에서는 앞서 정의한 보안 요구사항에 맞게 제안 기법이 설계되었음을 보인다. 5장 효율성 분석에서는 제안 기법 적용을 위해 기존 PKI 구조를 확장하였을 때 발생하는 저장 공간, 연산량, 통신량을 분석한다. 마지막으로 6장에서는 결론을 기술한다.

## II. 배경지식

### 2.1 공개키 기반 구조

공개키 기반 구조는 공개키 암호알고리즘을 안전하게 사용하기 위해 필요한 서비스를 제공하기 위한 기반 구조이다. PKI의 주된 목적은 사용자의 공개키와 소유자(subject)를 안전하게 연결해줌으로써 해당 공개키가 그 소유자의 것이라는 것을 확신할 수 있도록 해주는 것이다. 이는 PKI에서 인증서라는 전자 문서 형태로 관리되며 인증기관이 자신의 개인키를 사용하여 소유자의 인증서에 서명해줌으로써 해당 공개키가 그 소유자의 것이라는 것을 확신할 수 있게 해준다. 현재 가장 많이 사용되는 인증서 형식 표준은 ITU-T의 X.509[4]이며, 국내에서는 한국정보보호진흥원에서 국내 환경에 맞춰 정의한 X.509 v3 인증서 형식[15]를 사용하고 있다. 인증서에 들어가는 기본 항목으로는 인증기관, 일련번호, 인증서의 소유자, 유효기간, 공개키 등이 있으며, 이 기본 항목이 인증기관의 개인키로 서명되어 인증서에 포함되게 된다.

보안성 강화를 목적으로 PKI에서는 주기적으로 공개키를 갱신하도록 하고 있으며, 이는 인증서에 만료기간(expired date)으로 명시되어 있다. 이처럼 인증서 폐지 시점이 일정하게 정해져 있는 경우에는 폐지 시점을 인증서에 명시함으로써 자동적으로 폐지되도록 할 수 있다. 사용자의 소속이나 직위와 같은 상태정보가 인증서 기본 항목에 포함된 정보와 달라질 경우 이 시점부터 해당 인증서는 유효하지 않아야 한다. 인증서 만료 시점 전에 발생하는 사용자 상태정보 변동 시점은 정확히 예측하기 어렵기 때문에 별도의 폐지 방법을 필요로 한다. 사용자의 개인키가 분실, 훼손 또는 노출(compromise) 되었다고 판단되는 경우에도 개인키에 대응하는 인증서를 폐지해야 한다.

인증기관에서는 인증서 만료 시점 전에 인증서를 폐지할 수 있는 방법으로 공개키 기반 구조에서는 인증서 폐지 목록(CRL: Certificate Revocation List)[4]을 정의하여 공개 디렉토리에 게시하는 방법을 사용하고 있다. CRL 방식은 인증서 유효성을 검증하려는 검증자가 인증기관이 작성한 인증서 해지 목록을 디렉토리에서 주기적으로 다운로드 받아 인증서의 유효성을 검증하는 방식이다. 이 방식은 검증자가 주기적으로 최신 인증서 폐지 목록을 유지하기 위해 주기적으로 이를 다운로드 받아야 하는데 가입자 수가 증가할수록 인증서 폐지 목록의 사이즈가 커지는 문제점으로 인해 통신측면에서 비효율적인 방식이다. 또한 CRL을 이용한 인증서 상태 검증 방식은 인증서 상태에 대한 실시간성을 반영할 수 없기 때문에 인터넷뱅킹과 증권 트레이딩 등 실시간으로 신속하고 정확한 처리를 요구하는 환경에서는 적합하지 않다. 따라서 본 논문에서는 실시간성을 반영할 수 있는 OCSP를 이용한 인증서 유효성 검증 방식으로 제안 기법을 기술한다.

OCSP는 실시간으로 인증서의 유효성을 검증할 수 있는 프로토콜로써 OCSP를 이용한 인증서 검증과정은 Fig. 1.과 같다. 제안하는 기법은 서버-클라이언트 모델 상에서 서버가 사용자의 인증서를 검증해야 하는 구조이므로 Fig. 1.에서 서버는 사용자의 인증서 검증자의 역할을 한다. OCSP는 인증서 검증자가 특정 사용자의 인증서 상태를 OCSP 서버에 문의하면 OCSP 서버는 해당 인증서의 폐지 여부만을 검증자에게 알려줌으로써 검증자가 인증서 폐지 목록을 모두 다운로드 받아야 하는 부담을 줄인다. Fig.1.에서 인증기관은 CRL을 생성하여 디렉토리에 개시한다. OCSP 서버는 인증기관이 정한 CRL 갱신 정책에 따라 CRL을 지속적으로 갱신함으로써 최신 CRL 상태를 유지한다. 사용자 인증서의 유효성을 검증하려는 서버는 OCSP 서버로 사용자 인증서의 식별자를 포함한 유효성 질의 메시지(OCSP Request)를 전송한다. 이를 전송 받은 OCSP 서버는 인증서 식별자가 최신 CRL에 포함되어 있는지를 확인함으로써 유효성을 검사한 후, 그 결과를 서버로 응답(OCSP Response)해준다.

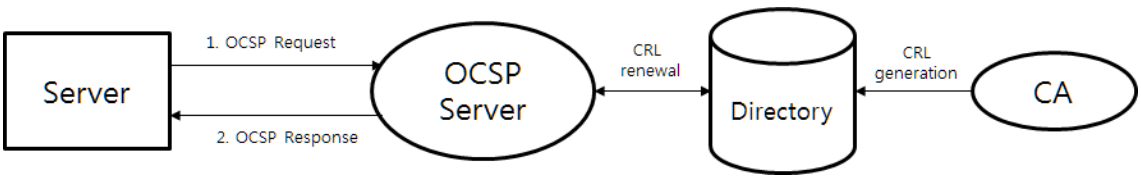


Fig. 1. Certificate verification process using OCSP

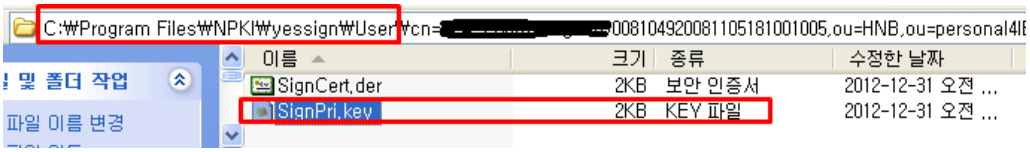


Fig. 2. Private key stored in HDD(Hard Disk Drive)

## 2.2 PKI 구조 문제 제기

PKI에서는 공인인증기관 간 상호연동성을 제공하기 위해 공개키 인증서와 개인키 파일을 표준화된 위치에 저장하도록 하고 있다[17]. 소프트 토큰 기반 공개키 인증서 및 개인키 파일의 저장위치는 Table 1.과 같다. Fig.2.에서 볼 수 있듯이 표준화된 저장 위치에는 사용자의 개인키가 파일형태로 존재하고 있어 개인키 파일 복제와 자동화 수집이 용이한 구조이다. 공인인증기관 간 상호연동성을 위해 표준화된 저장 위치는 필수적인 요소이지만 소프트 토큰 상에서 현재까지 사용자 개인키 유출에 대해서는 무방비 상태이다. 즉 소프트 토큰 기반 인증서에서 사용자 개인키 파일이 유출 되도 이를 탐지할 수 방법은 현재까지 존재하지 않는다.

파일 형태로 저장되어 있는 개인키를 보호하기 위하여 개인키 파일은 PBKDF2[13] 패스워드 기반 암호화 방식으로 암호화 되어 저장된다. 그러나 최근 GPU를 이용한 PBKDF2 공격 연구 결과[6]는 개인키 파일이 유출되면 사용자 개인키가 어렵지 않게 노출(compromise)될 수 있다는 위험성을 보여주고 있다.

현재의 PKI 구조에서는 사용자 개인키가 유출되거나 노출이 의심될 경우 사용자가 폐지 신청을 하도록 하고 있다. 그러나 파일 복제로 인한 개인키 파일 유출을 사용자가 인지하기는 어려우며, 패스워드 크랙으로 인해 유출된 개인키 파일로부터 개인키가 노출(compromise)될 경우 사용자 신원 도용이나 공격자에 의한 서명 값 생성을 차단할 수 있는 방법이 현재의 PKI 구조에서는 전무하다.

## 2.3 위협 모델

공개키 기반 구조에서 공개키/개인키 쌍은 핵심이 되는 요소이다. 그 중에서도 사용자의 서명용 개인키는 공개키 기반 구조에서 가장 중요한 요소로 꼽을 수 있다. 정당한 사용자의 개인키가 공격자에게 유출되면 공격자는 그 키를 이용하여 특정 전자 문서에 대한 사용자의 전자 서명 값을 생성할 수 있으며, 개인키 소유 증명 방법을 이용한 사용자 인증에 위장 공격을 시도할 수도 있다[7]. 사용자의 서명용 개인키 노출 위협은 다음과 같다[8].

- 알고리즘 공격(Algorithmic Attack): 서명 알고리즘 자체의 취약점을 이용하여 수학적 또는 암호학적 공격으로 서명 개인키를 계산하는 공격 방식을 의미하며 이에 대한 예로 Pollard[11]에 의해 Ong-Schnorr-Shamir 서명 기법[10]이 공격되었음.

Table 1. Storage location of soft-token digital certificate

| Storage medium | OS | | Storage location |
|---|---|---|---|
| HDD | Windows | Windows 98, ME, XP | (HDD volume name):\Program Files\NPKI\(DN of CA) |
| | | Windows vista and above | %UserProfile%\AppData\LocalLow\NPKI\(DN of CA) |
| | UNIX/Linux | | (User account)/NPKI/(DN of CA) |
| | Mac OS X | | (User account) Library/Preferences/NPKI/(DN of CA) |
| USB | Windows | | (Disk name):\NPKI\(DN of CA) |
| | UNIX/Linux | | (Mount directory)/NPKI/(DN of CA) |
| | Mac OS X | | /Volumes/(Disk name)/NPKI/(DN of CA) |

· 구현상의 오류(Implementation Failure): 특정 서명 알고리즘이 잘못 구현되었을 때 발생하는 취약점을 이용한 공격 방식이며, 취약 키(weak key) 선택이나 잘못된 난수 생성기 사용 또는 개인키가 충분히 보호되지 않은 경우가 이에 해당됨. 이에 대한 예로 ElGamal 서명 알고리즘이 잘못 구현된 경우에 대한 공격[3]이 있었음.
· 내부자 공격(Insider Attack): 이 공격은 서명 알고리즘이 사용될 때 서명키가 임시적으로 메모리에 올라가는 순간 이를 탈취하여 개인키를 얻어내거나 키 로깅(key logging)이나 사회 공학적(social engineering) 공격 방식으로 사용자가 복호화하기 위해 입력하는 패스워드를 노출하는 공격방식임.
· 전수조사 공격(Brute-force Attack): 패스워드 기반 방식으로 보호되어 있는 개인키를 노출하기 위해 패스워드 전수조사 공격이나 사전 공격(dictionary attack)[14,9,2,6]을 이용하여 개인키를 노출시키는 공격 방식임.

알고리즘 취약점이나 구현상의 오류로 인한 개인키 노출은 알고리즘 또는 소프트웨어 설계상 발생하는 오류이므로 이러한 위협은 연구범위를 벗어난다. 또한 내부자 공격과 같은 해킹에 의한 영구적인 시스템 장악에 의한 직접적인 패스워드 노출이나 사회 공학적 공격도 본 논문의 위협 범위에서 벗어난다. 제안하는 기법은 PKI 구조를 강화하기 위한 방식이며, 시스템 레벨에서의 해킹이나 사회 공학적 공격은 근본적으로 PKI 구조 자체에서는 다루기 힘든 주제이다. 이를 위한 보안 기법으로 안티바이러스 제품군, End-to-End 보안, 이중 인증(two-factor authentication)과 같은 방식이 더욱 효과적인 대응책일 것이다.

본 논문에서는 위에서 기술한 사용자 서명키 노출 위협 중 패스워드 전수조사 공격 위협에 초점을 두고 패스워드 전수조사 공격에 대하여 공격 시도를 탐지하여 유출된 인증서를 폐지시킬 수 있는 방법을 제안한다. 본 논문에서 다루는 공격 모델은 소프트 토큰 방식으로(패스워드 기반 암호화로 보호된) 저장된 사용자 개인키 파일을 수집하여 전수조사를 시도하는 공격자를 대상으로 한다. 공격자는 전수조사 공격, 사전 공격 등의 패스워드 공격[14]으로 사용자의 개인키를 보호하고 있는 패스워드를 계산할 때 까지 공격을 시도한다고 가정한다.

Weir 등[14]에 의해 제안된 패스워드 공격 알고리즘을 이용한 Kelley 등[9]의 실험 결과에 의하면 일반적으로 권고하는 8자리의 사용자 패스워드는 10억 회 정도의 추측 시도면 40.3% 확률로 패스워드 크랙이 가능하다. 또한 최근 연구 결과[2]로 하나의 GPU를 이용하여 MD5로 해시된 패스워드 추측 공격의 시도할 경우 초당 3억 회의 패스워드 추측 시도가 가능하다. 즉 3~4초 정도면 MD5로 해시된 8자리의 패스워드가 40.3% 확률로 노출된다는 결론을 얻을 수 있다. 특히 PBKDF2에 대해서 GPU를 이용하여 일주일 안에 65% 이상의 확률로 패스워드를 노출할 수 있다는 연구 결과[6]는 PBKDF2 방식으로 사용자 개인키를 보호하고 있는 패스워드도 충분히 공격될 수 있다는 것을 의미한다.

## III. 서명용 개인키 노출 탐지 기법

제안 기법의 주요 아이디어는 하나의 올바른 개인키와 구분 불가능한 $n-1$개의 위장 개인키를 소프트 토큰 기반의 사용자 저장 공간에 저장하게 함으로써 공격자가 사용자 개인키 파일을 유출하여 패스워드 크랙에 성공하더라도 올바른 개인키를 구분할 수 없게 하는 것이다. 또한 공격자가 위장 개인키를 사용하여 인증을 시도할 경우 OCSP 서버에서 이를 탐지하고 인증기관으로 폐지 요청을 하면 인증기관은 해당 인증서를 폐지하고 사용자에 통보할 수 있도록 하는 기법을 제안한다.

### 3.1 시스템 구성 및 보안 요구사항

제안하는 기법의 목적은 개인키의 노출로 인해 큰 피해가 발생할 수 있는 금융서비스와 같은 환경에서 개인키의 노출을 탐지하는 것이다. 기존의 PKI에서 개인키의 노출을 탐지하지 못하는 한계점을 개선하기 위해 시스템 구성과 추가적인 보안 요구사항을 제안하여 개인키의 노출을 탐지하고자 한다.

제안하는 시스템 모델은 Fig.3.과 같다. Fig.3.에서 사용자는 금융서비스와 같이 높은 보안 수준을 요구하는 서비스를 사용하려고 하는 객체이며, 서버는 은행과 같이 사용자에게 금융서비스를 제공하는 객체이다. 사용자와 서버는 상호 인증 및 보안 채널을 형성하기 위하여 SSL/TLS와 같은 공개키 기반 보안 프로토콜을 사용한다고 가정한다. 인증기관은 사용자와 서버 간 보안 채널 형성에 사용될 인증서를 발급해 주는 기관이이며 OCSP 서버는 인증기관에서 관리한다고 가정한다. 사용자는 위장 공개키를 포함한 인증

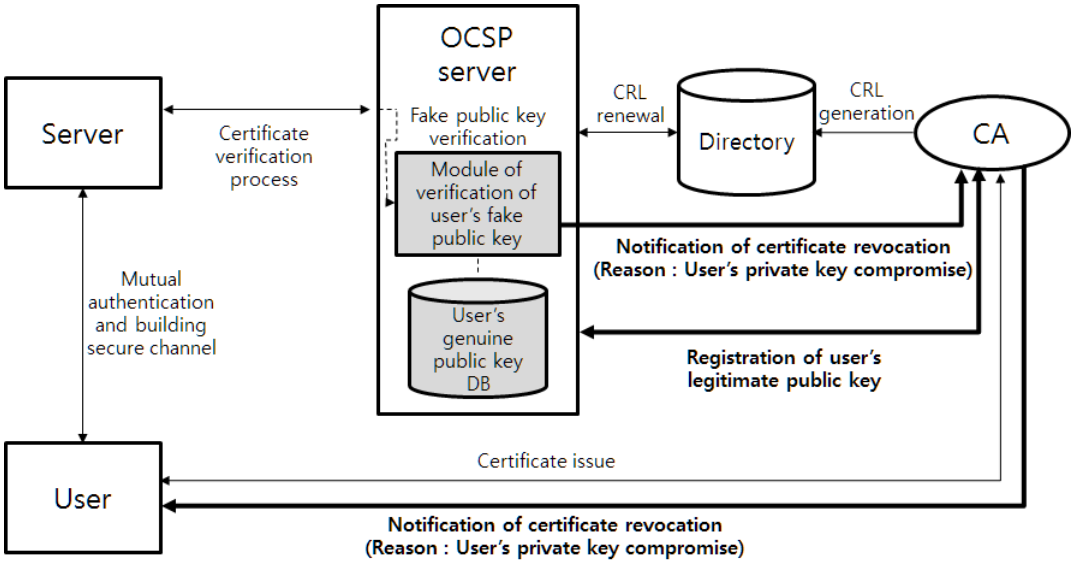


Fig. 3. System diagram for detection of private key compromise

서를 인증기관으로부터 발급받는다. 사용자 인증서 발급 절차에서 인증기관은 사용자의 올바른 공개키 식별자를 OCSP 서버에 등록함으로써 향후 사용자가 서버로 인증 요청을 할 때 사용된 사용자의 서명 값이 올바른 개인키로 생성된 값인지를 식별할 때 사용된다. 이 때 서명 값이 위장 개인키로 생성된 값이면 OCSP 서버에서는 사용자 서명키 유출 사유로 인증기관에 인증서 폐지 요청을 하고 서버로 유효하지 않은 인증서라는 응답을 한다.

사용자의 개인키가 유출되어 패스워드 공격이 시도되었는지를 탐지하기 위한 방법으로 제안하는 기법에서는 여러 개의 공개키/개인키 쌍을 사용할 수 있도록 설계하였다. $n$개의 공개키/개인키 쌍 중 올바른 값은 하나이며 나머지 $n-1$개는 공격자를 속이기 위한 위장 키 값으로 사용된다. 따라서 공격자가 패스워드 공격에 성공하였더라도 어떤 개인키가 올바른 값인지 구분할 수 없다면 공격자가 공격에 성공할 수 있는 확률은 $\frac{1}{n}$이 된다. 그러므로 공격자가 $n$개의 개인키 중 올바른 값을 구분할 수 없도록 설계되어야 한다. 패스워드 공격이 성공했다는 가정 하에 공격자가 올바른 개인키를 식별할 수 있는 위협요소와 이에 대한 보안 요구사항은 다음과 같다.

· 올바른 패스워드 식별: 올바른 패스워드와 $n-1$개의 위장 패스워드에서 공격자가 올바른 패스워드를 식별할 수 있는 확률이 $\frac{1}{n}$보다 커질 경우 제안 기법의 안전성은 떨어진다. 최악의 경우(worst case) 올바른 패스워드 식별 확률이 1일 경우 기존의 PKI 구조와 동일한 보안성을 갖게 된다. 따라서 공격자가 올바른 패스워드를 식별할 수 있는 확률이 $\frac{1}{n}$에 가까워질 수 있도록 위장 패스워드를 구성해야 한다.

· 올바른 공개키 값 노출: 사용자의 올바른 공개키 값이 노출 될 경우 공격자는 $k$개의 사용자 개인키로 서명 값을 생성한 후 올바른 공개키로 서명 값을 검증함으로써 올바른 개인키 값을 식별할 수 있다. 따라서 올바른 공개키 값은 서버와 인증기관 외의 제3자에게 노출되지 않아야 한다.

· 사용자 서명 값 노출: 사용자의 서명 값이 노출된다면 공격자는 $k$개의 사용자 공개키로 이를 검증함으로써 올바른 공개키를 식별할 수 있게 된다. 공격자가 올바른 공개키 값을 식별할 수 있게 되면 위의 올바른 공개키 값 노출에서와 같은 절차로 공격자는 올바른 개인키 값을 식별할 수 있게 된다. 따라서 사용자의 서명 값은 기밀로 유지되어야 한다.

금융서비스와 같은 환경에서 서명 값은 사용자 인증이나 부인방지 기능을 제공하기 위해 필요한 값으로써 서명 값은 서버에게만 전달되어 검증될 수 있으면

된다. 현재 금융서비스에서도 거래 정보 등에 대한 서명 값은 외부로 공개하지 않는다. 즉, 서명 값이 노출되지 않는 환경이므로 위의 요구사항은 현재의 PKI 환경에 적용 가능한 가정이다.

제안 기법에서 서버는 OCSP 서버를 통하여 사용자 공개키 값을 검증하는 과정에서 사용자의 올바른 공개키 값을 알 수 있다. 제안하는 시스템 모델에서의 서버는 금융기관과 같이 사용자 금융거래의 안전성을 보장해야 하는 의무가 있는 기관이므로 충분히 신뢰할 수 있는 객체(semi-trust or trust)라고 가정한다. 즉, 서버가 악의적으로 사용자의 올바른 공개키를 식별할 수 있는 정보를 외부로 노출시키지 않는다고 가정한다.

## 3.2 기법 설명

### 3.2.1 키 생성

우선 $n$개의 사용자 개인키 파일을 보호하기 위한 패스워드 설정 방법에 대해서 설명한다. 사용자의 패스워드 생성 시 Fig.4.와 같이 HEAD와 TAIL 부분으로 분류하여 패스워드를 입력하도록 한다. 여기서 사용자가 입력한 HEAD와 TAIL 부분을 붙이면 사용자가 실제로 사용하게 될 올바른 패스워드 $pw_1$가 된다. 여기서 TAIL 부분은 숫자로 입력하도록 한다. 올바른 패스워드 $pw_1$과 구분 불가능한 위장 패스워드를 생성할 때 TAIL 부분만을 동일한 자리 수의 난수 값을 선택하여 위장 패스워드 $pw_i (2 \le i \le n)$를 $n-1$개 생성한다. 이와 같이 TAIL 부분을 숫자로만 입력 받는 이유는 공격자가 올바른 패스워드와 위장 패스워드를 구분할 수 있는 확률을 $1/n$로 구성하기 위해서다. 예를 들어 123apple, 123appoe, 123appue와 같이 TAIL 부분을 문자열이 포함되게 구성한다면 공격자는 의미 없는 123appoe, 123appue를 배제하고 123apple과 같이 의미 있는 문자열이 포함된 패스워드를 올바른 패스워드로 구분할 수 있기 때문이다. TAIL 부분을 동일한 자리 수의 난수 값을 선택하게 하여 위장 패스워드를 생성함으로써 올바른 패스워드와 위장 패스워드를 공격자가 구분할 수 있는 확률을 $\frac{1}{n}$에 가깝게 만들어 줄 수 있다.

다음으로 $n$개의 공개키/개인키 쌍을 생성한다. 설명의 편의상 올바른 공개키/개인키를 $pk_1/sk_1$으로 표기하고 $pk_i/sk_i$ $(2 \le i \le n)$은 위장 공개키/개인키로 표기한다. 다음으로 위에서 생성한 패스워드로 PBKDF2 개인키 파일 암호화키를 생성한 후 각 개인키 파일을 암호화 한다.

향후 개인키로 서명 생성을 하려 할 때 개인키 파일을 복호 화할 수 있는 방법이 필요하다. 개인키 파일은 DER(Distinguished Encoding Rule) 인코딩 되어 있으므로 개인키 파일이 제대로 복호화 되었는지 아닌지를 DER 인코딩 형식을 확인함으로써 구분할 수 있다. 따라서 사용자가 올바른 패스워드를 입력하면 사용자 소프트웨어는 복호화가 제대로 되는 개인키 파일을 찾을 때까지 올바른 패스워드로 생성한 키로 복호화를 시도한다. 이 때 복호화가 제대로 되는 개인키 파일이 해당 패스워드와 대응되는 개인키가 된다.

### 3.2.2 인증서 발급 절차

기존 인증서 발급 절차[1,12,16]을 간략화 하면 Fig.5.와 같다. 인증서를 발급 받으려는 사용자는 우선 공개키/개인키 쌍을 생성한다. 다음으로 사용자 공개키를 포함한 인증서 발급 신청을 위한 사용자의 기본 정보인 CertificationRequestInfo를 생성한 후, 해당 공개키에 대한 개인키 소유증명을 위한 서명 값 $sig_{sk}\{M\}$을 생성하여 Certification -RequestInfo와 함께 인증기관으로 전송한다. 이를 전송 받은 인증기관은 사용자가 전송한 인증서 발급 신청 메시지에서 사용자 공개키를 추출한 후 사용자의 서명 값을 검증한다. 이 과정은 인가번호와 참조번호를 이용한 사용자 인증과정도 포함되어 있다. 인증기관은 사용자 인증과 서명 값 유효성을 확인되면 사용자 인증서 템플

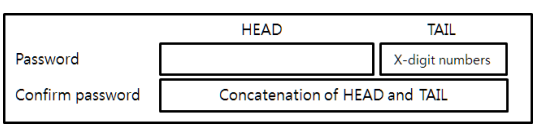


Fig. 4. Example of password generation interface

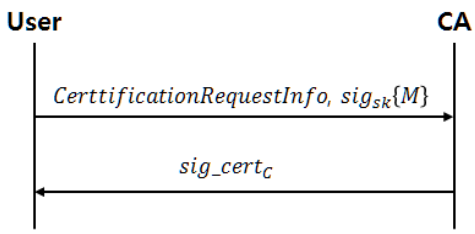


Fig. 5. Existing certificate issue protocol

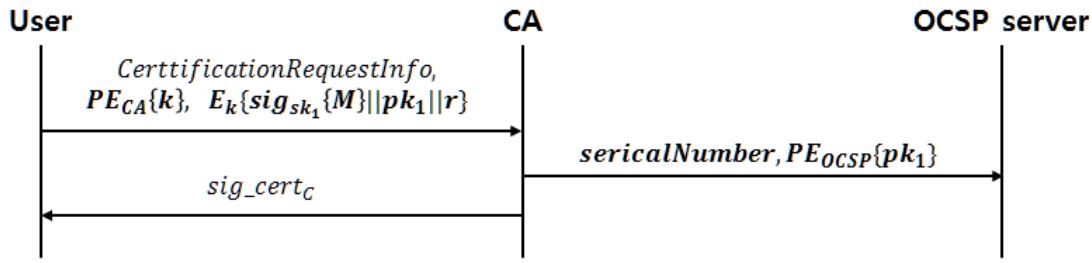


Fig. 6. Extension of certificate issue protocol

릿을 생성한 후 자신의 개인키로 서명하여 사용자 인증서를 생성하고 이를 사용자에게 전송함으로써 인증서 발급과정은 완료된다.

제안하는 공개키/개인키 구조를 기존 인증서 발급 프로토콜에 확장(Fig.6. 참고)하기 위해서는 사용자의 올바른 개인키 $sk_1$로 서명된 값 $sig_{sk_1}\{M\}$과 이를 검증하기 위한 올바른 공개키 $pk_1$는 기밀로 유지되어야 한다. 따라서 사용자는 랜덤하게 대칭키 $k$를 생성하여 서명 값 $sig_{sk_1}\{M\}$, 공개키 $pk_1$와 난수를 덧붙여 대칭키 $k$로 암호화 한다. 이 대칭키는 인증기관의 공개키로 암호화하여 전송한다.

향후 사용자의 개인키로 서명된 서명 값이 올바른 서명키로 생성된 값인지 검증하기 위해서 인증기관은 OCSP 서버에 올바른 서명 검증키인 $pk_1$을 안전하게 등록해야 한다. 인증기관은 사용자 인증서의 식별자인 $serialNumber$와 사용자의 올바른 공개키 $pk_1$을 OCSP 서버의 공개키로 암호화한 값 $PE_{OCSP}\{pk_1\}$을 OCSP 서버로 전송한다. 이를 전송 받은 OCSP 서버는 $serialNumber$와 해당 공개키 $pk_1$을 OCSP 서버의 사용자 공개키 DB에 안전하게 저장한다.

### 3.2.3 상호 인증 및 보안채널 형성

제안 기법 응용에 대한 예로 사용자와 서버 간 상호 인증과 보안채널 형성 기능을 제공하는 기법 중 가장 널리 쓰이고 있는 규격인 SSL/TLS[5]를 확장하여 제안 기법을 적용할 수 있는 방법에 대해서 설명한다.

기존 SSL/TLS 핸드쉐이크 프로토콜 중 RSA 키 교환 방법을 이용한 서버와 사용자간 인증 과정을 간략화 하면 Fig.7.과 같다. 서버가 RSA 암호화용 인증서 Enc_Cert 를 사용자에게 전송하면 사용자는 서버의 인증서 유효성을 검증한 후 세션 키 생성에 사용할 $premaster$를 서버의 공개키로 암호화 값 $PE_s\{premaster\}$, 사용자의 RSA 서명용 인증서 Sig_Cert, 자신의 개인키 소유를 서버에게 입증하기 위해 이전까지 핸드쉐이크 메시지 $M$을 서명한 값 $sig_{sk}\{M\}$을 서버로 전송한다. 이를 수신한 서버는 사용자 인증서의 유효성을 검증한 후 OCSP로부터 인증서 폐지 상태를 확인 요청을 하고 상태 정보를 응답받는다. 다음으로 사용자 서명 값을 인증서에 포함되어 있는 공개키로 검증함으로써 사용자를 인증하고 자신의 개인키로 $premaster$를 복호화하여 세션 키를 생성한다. 향후 사용자는 세션 키 검증과정에서 서버가 올바른 세션 키 값을 생성하면 서버를 인증하게 된다.

제안 기법을 SSL/TLS 핸드쉐이크 상호인증 과정

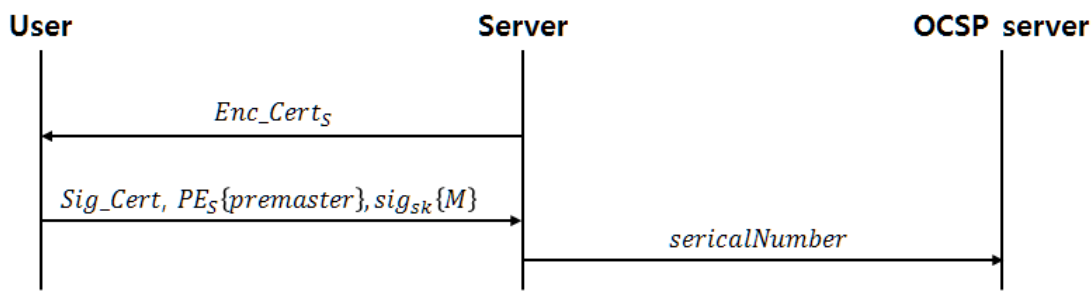


Fig. 7. Existing mutual authentication process in SSL/TLS handshake

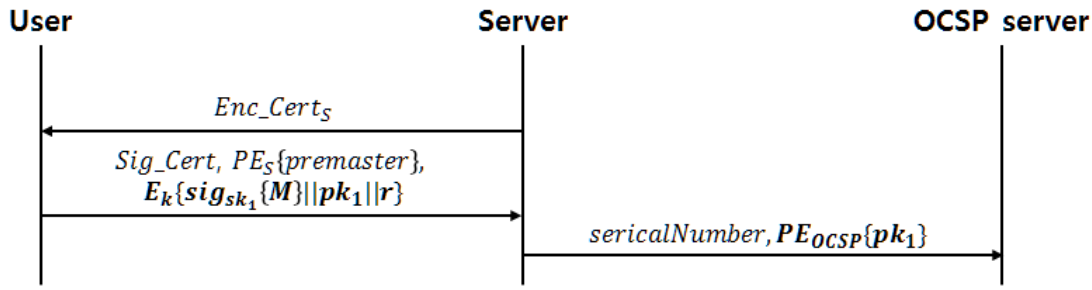


Fig. 8. Extension of mutual authentication process in SSL/TLS handshake

에 적용하기 위해서는 사용자의 올바른 개인키 $sk_1$로 서명된 서명 값 $sig_{sk_1}\{M\}$와 서명 값을 검증할 올바른 공개키 $pk_1$의 기밀성이 보장되어야 한다. 제안 기법을 SSL/TLS로 확장한 기법은 Fig.8.과 같다. SSL/TLS ServerHelloDone 메시지 이후에 사용자는 $premaster$ 값과 ClientHello, ServerHello 메시지에서 생성한 서버의 난수와 사용자의 난수로 세션 키 $k$를 생성할 수 있는 상태가 된다. 사용자는 사용자의 서명 값 $sig_c\{M\}$, 올바른 사용자 공개키 $pk_1$과 난수 $r$을 덧붙여 세션 키 $k$로 암호화하여 서버로 전송한다.

### 3.2.4 개인키 노출 탐지 및 인증서 폐지

앞서 설명한 SSL/TLS 핸드쉐이크 과정에서 사용자의 인증서를 전송받은 서버는 사용자 인증서 폐지 여부를 확인하기 위해 OCSP 서버로 사용자 인증서 폐지 확인 요청을 한다. 이 과정에서 OCSP 서버는 사용자의 인증서 폐지 목록을 확인함으로써 인증서 유효성을 확인한다. 제안하는 사용자 개인키 노출 탐지는 이 과정에서 OCSP 서버에 의해 이루어진다. 개인키 노출 탐지 및 인증서 폐지를 위한 OCSP 검증과정은 Fig.9.와 같다.

서버는 사용자의 인증서 유효성 확인 요청 메시지로써 사용자 인증서 식별자 $serialNumber$, 공개키 $pk_i$를 OCSP 서버의 공개키로 암호화한 값 $PE_{OCSP}\{pk_i\}$를 OCSP 서버로 전송한다. 이를 수신한 OCSP 서버는 인증서 폐지 목록에 사용자의 인증서가 포함되어 있는지를 $serialNumber$로 검색한 후 이를 통과하면 위장 공개키 검증 단계로 넘어간다. OCSP 서버는 $serialNumber$에 대응하는 사용자 서명 검증키 $pk_i$가 사용자 공개키 DB에 포함되어 있으면 정상 상태로 서버에 응답을 한다. 만약 사용자 공개키 DB에 저장되어 있는 $pk_1$과 서버로부터 전송 받

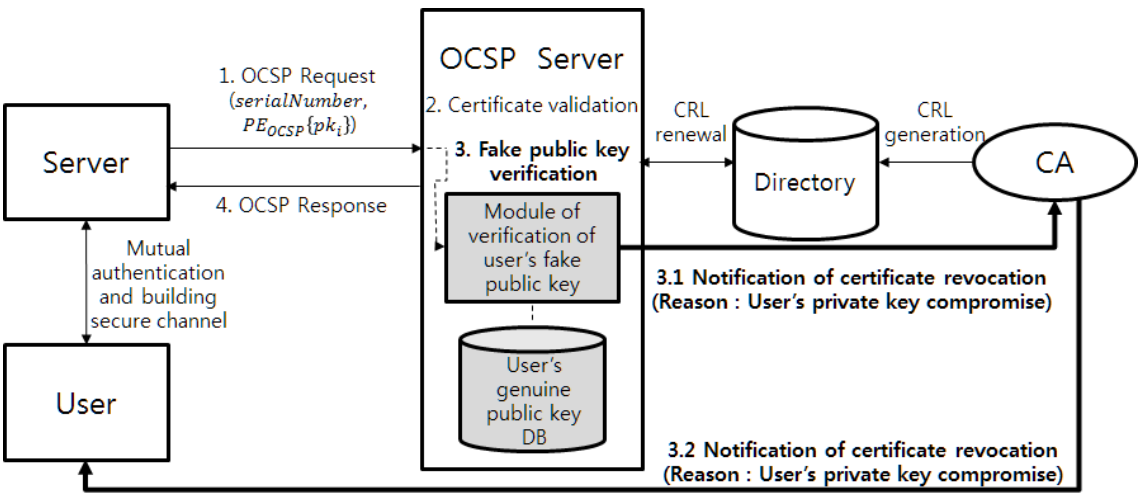


Fig. 9. Process for detection of private key compromise and notification of certificate revocation

은 $pk_i$에 대응하는 공개키 지수 식별자가 일치하지 않으면 사용자 인증서가 유출되어 패스워드 크랙이 시도된 것으로 탐지한다. 사용자 개인키 노출이 탐지되면 OCSP 서버는 즉시 인증기관에 "사용자 서명키 유출" 사유로 해당 사용자 인증서 폐지 요청을 한다. 이를 전송 받은 인증기관은 해당 인증서를 즉시 인증서 폐지 목록에 포함시키고 사용자에게 "사용자 서명키 유출"을 사유로 인증서를 폐지했다는 통보를 한다.

OCSP를 이용한 위장 공개키 검증 단계는 최초 1회만 수행하고 서버는 사용자의 올바른 공개키 값 식별정보를 안전하게 보관한다. 이후부터 서버는 OCSP를 이용한 추가 연산 없이도 보관한 사용자의 올바른 공개키 값 식별정보를 이용하여 사용자의 올바른 개인키 사용 여부를 판단할 수 있게 된다.

## IV. 안정성 분석

### 4.1 패스워드 공격에 대한 안전성

· 어떤 PBKDF2 크랙 알고리즘이 $p(0 \le p \le 1)$의 확률로 $pLen$ 길이의 패스워드를 찾는데 걸리는 평균 시간을 $crackT_{pLen}$라고 하고, 사용자의 개인키 파일이 $pLen$ 길이의 패스워드로 보호되어 있다고 가정하자. 사용자의 개인키 파일이 유출되어도 공격자는 평균 $crackT_{pLen}$ 시간 동안에 이 크랙 알고리즘으로 올바른 서명 값을 생성할 수 있는 확률은 $\frac{p}{n}$이다.

기존 PBKDF2 기반으로 보호된 개인키 파일이 유출되었을 때 공격자가 $crackT_{pLen}$ 시간으로 올바른 서명 값을 생성할 수 있는 확률은 $p$이다. 즉, 현재의 개인키 파일 구조에서는 공격자가 PBKDF2 기반 패스워드 크랙에 성공했다는 조건하에 올바른 서명 값을 생성할 수 있는 확률은 1이 된다. 다음으로 제안하는 개인키 파일 생성 방식으로 개인키 파일이 생성했을 때 이 개인키 파일이 공격자에게 유출 되었다고 가정하자. 그러면 공격자는 $crackT_{pLen}$ 시간에 $p$의 확률로 크랙에 성공할 수 있다. 이 때 공격자는 $n$개의 개인키 중 올바른 개인키를 식별할 수 있어야 올바른 서명을 생성할 수 있다. 개인키는 모두 난수 값이므로 공격자가 올바른 개인키 지수를 선택할 수 있는 확률은 $\frac{1}{n}$이다. 따라서 제안기법으로 생성된 개인키 파일이 유출되었을 때 공격자가 올바른 서명 값을 생성할 수 있는 확률은 $\frac{p}{n}(n>1)$이다.

· 제안 기법의 사용자 개인키 파일이 유출되어 공격자가 PBKDF2 크랙에 성공했다고 하더라도 OCSP 서버에서 이를 탐지할 수 있는 확률은 $1-\frac{1}{n}$이다.

공격자가 유출된 개인키 파일에 대해서 PBKDF2 크랙에 성공했다고 하더라도 올바른 서명키를 선택할 확률은 $\frac{1}{n}$이다. 따라서 OCSP 서버에서는 $1-\frac{1}{n}$ 확률로 개인키 파일 유출을 탐지하여 해당 인증서를 폐지하고 사용자에게 이를 통보할 수 있다.

### 4.2 올바른 개인키 구분의 어려움

제안 기법으로 생성된 개인키 파일이 유출되었을 때, 공격자가 $\frac{p}{n}$보다 높은 확률로 올바른 서명 값을 생성하기 위해서는 올바른 개인키를 구분할 수 있어야 한다. 공격자에게 올바른 개인키 식별 정보가 노출될 수 있는 경우는 3.1장에서 언급한 올바른 공개키 식별자가 노출되거나 사용자의 올바른 서명 값이 노출되는 경우이다. 사용자의 올바른 공개키 식별자가 유출될 수 있는 구간은 다음과 같다.

· 인증서 발급 요청 과정에서 사용자가 올바른 공개키를 세션 키로 암호화하여 인증기관으로 전송하는 과정
· 인증서 발급 요청 과정에서 인증기관이 OCSP로 사용자의 올바른 공개키를 OCSP 서버의 공개키로 암호화하여 전송하는 과정
· SSL/TLS에서 사용자가 공개키를 세션 키로 암호화하여 서버로 전송하는 과정
· SSL/TLS에서 서버가 사용자의 공개키 검증을 위하여 OCSP 서버로 사용자 공개키를 OCSP 서버의 공개키로 암호화하여 전송하는 과정

다음으로 사용자의 서명 값이 유출될 수 있는 구간은 다음과 같다.

· 인증서 발급 요청 과정에서 사용자의 개인키 소유 증명 과정에서의 사용자 서명

Table 2. Terminology of efficiency analysis of a proposed method

| Terminology | Explanation |
|---|---|
| $n$ | The number of private / public key pairs |
| $B_{PK}$ | Public key length |
| $B_{privM}$ | Size of private key file |
| $B_{sn}$ | Length of certificate serial number |
| $B_{PE}$ | Length of ciphertext or signature in public key cryptosystem |
| $B_r$ | Length of random number $r$ |
| $N_{user}$ | The number of PKI members |
| $S$ | A operation of symmetric cryptosystem |
| $P$ | A operation of asymmetric cryptosystem |
| $KeySearch$ | A operation for finding legitimate key among $n$-encrypted private keys<br>In the worst case, $n$-decryption operations are required. |

· SSL/TLS에서 사용자 인증을 위한 개인키 소유 증명 과정

사용자의 올바른 공개키와 서명 값은 모두 전송 대상자의 공개키로 암호화 되어 있거나 대칭키로 암호화 되어 있다. 이 때 사용되는 공개키 암호 알고리즘과 대칭키 암호 알고리즘이 안전하다면 공격자는 암호화 되어 있는 값으로 어떠한 정보도 알아낼 수 없다. 따라서 메시지 전송 과정에서 올바른 공개키 식별 정보와 서명 값은 노출되지 않는다.

## V. 효율성 분석

본 장에서는 제안하는 개인키 유출 탐지 기법을 기존 PKI 시스템에 적용할 경우 추가적으로 요구되는 저장 공간, 연산량, 통신량을 분석한다. Table 2.는 분석에 사용할 표기법을 정리한 표이다.

### 5.1 저장 공간

제안 기법을 적용할 경우 기존 PKI 시스템과 비교하여 추가적으로 요구되는 사용자 저장 공간과 OCSP 서버의 저장 공간은 Table 3.과 같다. 사용자 측면에서 하나의 공개키 인증서 안에는 하나의 올바른 공개키와 $n-1$개의 위장 공개키가 저장되므로 $n-1$개의 위장 공개키 저장 공간이 추가적으로 요구된다. 또한 각 공개키에 대응되는 개인키 파일은 $n$개이므로 $n-1$개의 위장 개인키 파일 저장을 위한 추가 저장 공간이 요구된다. OCSP 서버에서는 위장 공개키 검증을 위해 각 사용자 별로 인증서 식별자인 $serialNumber$와 그에 대응되는 사용자의 올바른 공개키를 저장 공간이 추가적으로 요구된다.

Table 3. Additional storage space for a proposed method

| | User | OCSP server |
|---|---|---|
| Addtional storage space | $(n-1)(B_{PK}+B_{privM})$ | $N_{user}(B_{sn}+B_{PK})$ |

하나의 개인키 파일은 2 kB 미만이며 하나의 공개키 크기는 RSA 2048 비트 기준 512 바이트 미만이다. 10개의 공개키/개인키 쌍을 생성했을 경우 개인키 파일이 유출되어 패스워드 크랙이 되더라도 공격자의 공격 성공 확률은 10%가 된다. 이 때 요구되는 사용자의 저장 공간은 30 kB로 현재의 사용자 디바이스에서는 극히 미미한 저장 공간이다. 100개의 공개

Table 4. Additional operation and traffic in the process of certificate issue

| | User | | CA | | OCSP server |
|---|---|---|---|---|---|
| Operation | $P+S$ | | $2P+S$ | | $P$ |
| Transmission traffic | CA | $B_{PE}+B_{PK}+B_r$ | OCSP server | $B_{sn}+B_{PE}$ | - |
| | | | User | $(n-1)B_{PK}$ | |

Table 5. Additional operation and traffic in SSL/TLS and OCSP

| | User | | CA | | OCSP server |
|---|---|---|---|---|---|
| Operation | $S+KeySearch$ | | $P+S$ | | $P$ |
| Transmission traffic | Server | $(n-1)B_{PK}$ $+B_{PK}+B_r$ | OCSP server | $B_{PE}$ | - |

키/개인키 쌍일 경우 공격 성공 확률은 1%이며(유출 탐지율 99%), 추가 저장 공간은 300 kB이다. 즉 제안 기법은 소프트 기반으로 저장되어 있는 개인키 파일의 보안 강도와 사용자 저장 공간을 트레이드오프(trade off)하여 적용할 수 있는 구조이다.

### 5.2 연산량 및 통신량

이번 절에서는 인증서 발급 과정과 SSL/TLS, OCSP 과정에서 발생되는 추가 연산량과 통신량에 대해서 분석한다. 통신량은 전송량과 수신량을 분류하여 표기하였다. Table 4.는 인증서 발급 과정에서 요구되는 추가 연산량과 통신량이며, Table 5.는 SSL/TLS, OCSP에서 요구되는 추가 연산량과 통신량이다. 연산량은 공통적으로 각 공개키 연산 1~2회 ,대칭키 연산 1회 정도가 추가되므로 필요로 하는 추가 연산량은 극히 적음을 볼 수 있다.

통신량의 경우 사용자가 $n$개의 공개키가 포함되어 있는 인증서를 전송 할 때를 제외하고는 극히 적음을 볼 수 있다. 인증서 전송량은 앞서 저장 공간에서 수치로 예를 들어 설명한 것과 같이 제안 기법의 안전성 측면과 트레이드오프 할 수 있는 부분이다.

## VI. 결 론

본 논문에서는 PKI에서의 표준화된 개인키 저장 위치로 인해 발생할 수 있는 개인키 노출 문제에 대해서 다루었으며, 개인키 유출 시 공격자가 개인키를 보호하고 있는 패스워드를 크랙 하더라도 공격 성공확률을 기존 1에서 $\frac{1}{n}$로 낮출 수 있는 기법을 제안하였다. 또한 추가 인프라 구축비용 없이 기존 PKI 인프라를 확장하여 개인키 파일의 보안 강도를 높였으며, 유출된 개인키 파일이 공격자에게 노출되더라도 $1-\frac{1}{n}$확률로 개인키 유출을 탐지하고 그에 대응되는 인증서를 인증기관에서 폐지할 수 있게 하는 새로운 개념을 소개하였다. 마지막으로 제안 기법의 안전성을 분석함으로써 기존 PKI 보다 보안강도가 높아짐을 입증하였으며, 제안 기법을 기존 PKI 구조에 확장할 경우에 대한 추가 저장 공간, 연산량, 통신량을 분석함으로써 기존 PKI에 실제 적용하려 할 때 보안강도와 효율성을 트레이드오프 할 수 있도록 하였다.

## References


[1] C. Adams, S. Farrell, T. Kause, and T. Mononen, "Internet X.509 Public Key Infrastructure Certificate Management Protocol CMP," RFC 4210, Sep. 2005.

[2] M. Bakker and R. van der Jagt, "GPU-based password cracking," Univ. of Amsterdam, Feb. 2010.

[3] D. Bleichenbacher, "GeneratingElGamal Signatures Without Knowing the Secret Key," Advances in Cryptology, Eurocrypt'96, LNCS 1070,pp. 10-18, 1996.

[4] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "internet x.509 public key infrastructure certificate and certificate revocation list(CRL)profile." RFC 5280, May 2008.

[5] T. Dierks and E. Rescorla. "the transport layer security(TLS) protocol version 1.2," RFC 5246, Aug. 2008.

[6] M. Dürmuth, T. Güneysu, M. Kasper, C. Paar, T. Yalçin and R. Zimmermann, "Evaluation of Standardized Password-Based Key Derivation against Parallel Processing Platforms," ESORICS 2012, LNCS 7459, pp. 716-733, 2012.

[7] C. Ellison and B. Schneier, "Ten risks of PKI: What You're Not Being Told About Public Key Infrastructure,"Computer Security Journal, Vol. 16, No. 1, pp. 1-7,

2000.
〔8〕 M. Just and P. Oorschot, "Addressing the problem of undetected signature key compromise," Proceedings of the Network and Distributed System Security Symposium(NDSS), 1999.
〔9〕 P. Kelley, S. Komanduri, M. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christic, L.F. Carnor, and J. Lopez, "Guess again (and again and again): Measuring password strength by simulating password cracking algorithm," IEEE Symposium on Security and Privacy(S&P), pp. 523-537, May 2012.
〔10〕 H. Ong, C. Schnorr and, A. Shamir, "An Efficient Signature Scheme Based on Quadratic Equations," Proceedings of the 16th Annual ACM Symposium on Theory of Computing, pp. 208-216, 1984.
〔11〕 J. Pollard and C. Schnorr, "An Efficient Solution of the Congruence $x^2+ky^2=m \pmod n$," IEEE Transactions on Information Theory, Vol. 33, no. 5, pp. 702-709, 1987.
〔12〕 RSA Laboratories, "Certification Request Syntax Standard," PKCS#10, May 2000.
〔13〕 RSA Laboratories, "Password-Based Cryptography Standard," PKCS#5, Oct. 2012.
〔14〕 M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek, "Password cracking using probabilistic context-free grammars," IEEE Symposium on Security and Privacy(S&P), pp. 162-175, May 2009.
〔15〕 KISA, "Digital Signature Certificate Profile," KCAC.TS.CERTPROF, Sep. 2009.
〔16〕 KISA, "Accredited Certificate Request Message Format Specification," KCAC.TS.CRMF, Sep. 2009.
〔17〕 KISA, "User Interface Specification for the Interoperability between Accredited Certification Authorities," KCAC.TS.UI, Dec. 2012.
〔18〕 KISA, "Research on the Actual condition of Electronic Signature System usage(in Electronic Signature User)," KISA-WP-2012-0003, Dec. 2011.
〔19〕 Ministry of Security and Public Administration, "Accredited certificate user statistics," Jun. 2012.

## 〈저 자 소 개〉

박 문 찬 (Moon-chan Park) 학생회원
2013년 2월: 서울시립대학교 수학과 학사
2014년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 소프트웨어 분석, 소프트웨어 난독화

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: 교려대학교 정보보호대학원 교수
〈관심분야〉 암호프로토콜, 암호이론, USN이론, 키 교환, 익명성 연구, PET기술