情報保護學會論文誌
第17卷 第1號, 2007. 2

삭제된 공인인증서의 복구 및 개인키 암호화 패스워드의 검출*

최윤성[1], 이영교[1], 이윤호[1], 박상준[1], 양형규[2], 김승주[1], 원동호[1†]
[1]성균관대학교 정보통신공학부 정보보호연구소, [2]강남대학교 컴퓨터미디어공학부

The Recovery of the Deleted Certificate and the Detection of the Private-Key Encryption Password

Youn-sung Choi[1], Young-gyo Lee[1], Yun-ho Lee[1], Sang-joon Park[1],
Hyung-kyu Yang[2], Seung-joo Kim[1], Dong-ho Won[1†]
[1]Information Security Group, School of Information and Communication Engineering,Sungkyunkwan University
[2]Department of Computer & Media Engineering, Kangnam University

요 약

공인인증서는 온라인 금융거래와 증권거래 등에서 사용자의 신원 확인을 위해 사용된다. 이때 사용자의 공개키는 공인인증서에 저장되며, 이 공개키에 대응되는 사용자의 개인키는 보안을 위해 사용자가 설정하는 패스워드로 암호화되어 개인키 저장 파일에 저장된다. 본 논문에서는 현재 널리 사용되고 있는 공인인증서 관리 소프트웨어에서 정상적으로 삭제된 공인인증서와 개인키 저장 파일이 포렌식 툴을 이용하면 얼마든지 복구가 가능하다는 점을 밝힌다. 그리고 복구된 공인인증서와 개인키 저장 파일을 이용하여 오프라인에서 개인키 암호화 패스워드를 밝혀낼 수 있다는 문제점을 지적하고 그에 따른 대응책을 제시한다.

ABSTRACT

The certificate is used to confirm and prove the user's identity in online finance and stocks business. A user's public key is stored in the certificate (for e.g., SignCert.der) and the private key, corresponding to public key, is stored in the private key file (for e.g., SignPri.key) after encryption using the password that he/she created for security. In this paper, we show that the certificate, deleted by the commercial certificate software, can be recovered without limitation using the commercial forensic tools. In addition, we explain the problem that the private key encryption password can be detected using the SignCert.der and the SignPri.key in off-line and propose the countermeasure about the problem.

Keywords : *Digital Forensic, Data Recovery, PKI*

접수일: 2006년 10월 10일; 채택일: 2006년 11월 28일
* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음. (IITA-2006-C1090-0603-0028)
† 교신저자, dhwon@security.re.kr

Ⅰ. 서 론

정보통신을 위한 기술이 발달함에 따라 인터넷을 통한 전자상거래나 금융거래가 일반화되고 있다. 이러한 인터넷 서비스를 이용할 때 전송 데이터가 평문 그대로 전송된다면, 위조나 변조 같은 문제가 발생할 수 있다. 이러한 문제를 해결하기 위한 방법으로 공개키 기반의 전자서명이 사용되고 있다, 이 전자서명시스템에서 공인인증서는 전자상거래시 사용자의 신원을 확인하고, 문서의 위조와 변조, 거래 사실의 부인 방지 등을 목적으로 공인인증기관(CA)이 발행하는 전자적 정보로서, 일종의 사이버 거래용 인감증명서 역할을 한다[4].

전자서명시스템에서 사용되는 공인인증서와 개인키 저장 파일은 현재 X.509 v3의 기준에 따라서 작성되며, BER/DER 방식으로 인코딩되어 저장된다. 공인인증서에는 사용자 공개키가 저장되며 개인키 저장 파일에는 사용자 개인키가 저장되는데, 사용자의 개인키는 다른 사용자에게 노출되면 보안상의 위험이 있으므로 SEED 블록 암호 알고리즘을 이용하여 암호화 된다. SEED 블록 암호 알고리즘에 사용되는 비밀키는 사용자의 개인키 암호화 패스워드를 이용하여 생성된다[6].

공인인증서 시스템의 안전한 사용을 위해서 공인인증서와 개인키 저장 파일에는 암호화 기술이 적용되어 위/변조가 어려울 뿐만 아니라, 사용자가 설정하는 8 자리 이상의 개인키 암호화 패스워드가 사용되어 안정성을 높인다. 공인인증서는 개인정보나 카드결제정보, 계좌번호 등 인터넷에서 주고받는 정보에 대하여 신뢰성을 보장해준다. 그리고 개인 식별 수단으로도 사용되어 주민등록번호와 같은 개인정보를 공인인증서로 대체하고 있다.

현재 공인인증서를 관리하는 상용 소프트웨어들의 기능에는 사용자의 개인키 암호화 패스워드를 입력받아 확인하는 기능뿐만 아니라, 공인인증서 저장매체의 변경, 인증서 암호(개인키 암호화 패스워드) 변경, 인증서 보기/검증과 인증서 삭제 등이 있다. 이때 공인인증서 관리 소프트웨어에 의해서 정상적으로 삭제된 공인인증서가 복구가 가능하여 자신의 공인인증서 파일과 개인키 저장 파일이 어떠한 방법으로든 다른 사람에게 노출되어서는 안 된다. 더욱이 다른 사용자가 정상적인 사용자가 설정한 개인키 암호화 패스워드까지 알아낼 수 있다면 심각한 문제가 아닐 수 없다.

본 논문에서는 2장에서 현재 공인인증서 관리 소프트웨어에 대하여 살펴본 후, 공인인증서와 개인키 저장 파일에 저장된 정보의 프로파일을 알아본다. 그리고 사용자의 개인키 암호화 패스워드를 이용하여 SEED 블록 암호 알고리즘에 사용되는 비밀키와 초기벡터를 생성하는 방식을 살펴본다. 그리고 3장에서 상용 공인인증서 소프트웨어 상에서 정상적으로 삭제된 공인인증서와 개인키 저장 파일이 완전히 삭제되어야 함에도 불구하고 포렌식 툴을 이용하면 아무런 제약 없이 복구될 수 있음을 지적한다. 4장에서는 복구된 공인인증서와 개인키 파일을 이용하여 오프라인 상에서 사용자의 개인키 암호화 패스워드를 검출하는 방법을 설명한다. 5장에서는 검출된 개인키 암호화 패스워드를 검증하는 과정에 대해서 살펴본다. 그리고 6장에서 구현된 개인키 암호화 패스워드 검출/검증 프로그램의 성능을 평가한다. 7장에서는 제시한 문제점에 대한 해결방안을 제안하고, 8장에서 결론을 맺는다.

Ⅱ. 공인인증서 시스템의 공개키/개인키 관리체계

본 장에서는 한국 공인인증서 시스템에서 사용되는 공인인증서 관리 소프트웨어 및 공인인증서와 개인키 저장 파일의 프로파일을 알아보고, 공인인증서 소프트웨어에서 사용자가 설정하는 개인키 암호화 패스워드를 이용하여 생성한 비밀키와 SEED 블록 암호 알고리즘으로 사용자의 개인키를 암호화 시키는 기술에 대해서 살펴본다.

2.1 상용 공인인증서 소프트웨어

공인인증서를 관리하는 상용 소프트웨어들의 기능에는 공인인증서 저장매체의 변경, 인증서 암호(개인키 암호화 패스워드)의 변경, 인증서 보기/검증과 인증서 삭제 등이 있다. 하지만 소프트웨어들 종류와 이용하는 방식에 따라서 관리 방식이 조금씩 차이가 있다. 〔그림 1〕은 비 은행권에서 사용하는 공인인증서 소프트웨어들의 예이다.

〔그림 1〕의 (가)은 대한민국전자정부 홈페이지에서 주민등록등본 발급과 같은 서비스를 제공할 때, 사용자의 신원을 확인하기 위해서 공인인증서를 요구하는 소프트웨어 실행화면이다. (나)는 증권회사 홈페이지에서 사용자에게 증권거래 서비스를 제공할 때, 사용자의 신원을 확인하기 위해서 공인인증서를 요구하는 소프트웨

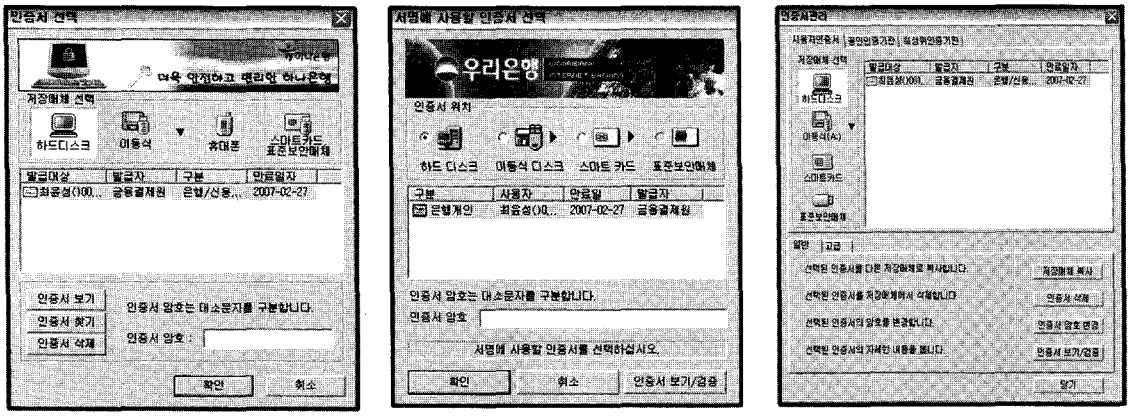
어 실행화면의 예이다. (다)는 인터넷 쇼핑몰에서 물품을 선택한 후, 신용카드 결제나 실시간 계좌이체를 이용하여 결제를 할 때 공인인증서를 요구하는 소프트웨어 실행화면이다. (나)와 같은 경우에는 홈페이지에서 공인인증서 발급/삭제 및 폐기, 인증서 비밀번호(개인키 암호화 패스워드) 변경과 같은 공인인증서 관리를 지원한다. (가),(다)와 같은 경우 공인인증서를 이용한 신원확인을 지원하고, 공인인증서 삭제와 같은 공인인증서 관리는 지원하지 않는다. 〔그림 2〕는 은행권에서 사용하는 공인인증서 소프트웨어들의 예이다.

〔그림 2〕의 (가), (나)는 은행 홈페이지에서 금융 서비스를 제공할 때, 사용자에게 공인인증서를 요구하는 소프트웨어 실행화면이다. (가)와 같은 경우에는 신원확인 기능뿐만 아니라 인증서 보기와 인증서 찾기, 인증서 삭제 등과 같은 공인인증서의 기본적인 관리 기능을 제공하는 반면, (나)은 신원확인만을 지원한다. (다)는 (가)과 (나)와 함께 실행되는 공인인증서 관리 소프트웨어로, 사용자에게 인증서 삭제, 인증서 암호(개인키 암호화 패스워드) 변경, 인증서 보기/검증과 같은 기능을 제공한다.

자신의 공인인증서 파일과 개인키 저장 파일은 어떠한 방법으로든 다른 사람에게 노출되어서는 안 된다. 그리므로 현재 사용하고 있는 컴퓨터에 저장된 공인인증서 파일과 개인키 저장 파일이 더 이상 현재 컴퓨터에서 사용하지 않을 경우에는 쉽게 삭제할 수 있어야 한다. 그러나 〔그림 1〕의 (가), (다)의 경우에는 해당 소프트웨어에서 공인인증서를 삭제하는 기능하지 않으며, 해당 홈페이지에서도 삭제 기능을 제공하지 않는다.

(가) (나) (다)

〔그림 1〕 공인인증서 관리 소프트웨어 - 비은행권



(가) (나) (다)

〔그림 2〕 공인인증서 관리 소프트웨어 - 은행권

[그림 1]의 (나)와 같은 경우에는 해당 소프트웨어에서 공인인증서를 삭제하는 기능은 제공하지 않으며, 해당 홈페이지에서만 공인인증서를 관리할 수 있게 되어 있어서 삭제 기능을 사용하기 위해서는 홈페이지의 공인인증서 관리 페이지로 이동하여야만 한다. [그림 1]의 (나)와 같은 경우에는 해당 소프트웨어에서 공인인증서를 삭제하는 기능은 제공하지 않지만 함께 실행되는 (다)과 같은 소프트웨어에서 삭제기능을 제공할 뿐만 아니라, 해당 홈페이지에서 삭제기능을 제공한다. (나)와 같은 경우에는 해당 소프트웨어에서 삭제기능을 제공할 뿐만 아니라, (다)와 같은 소프트웨어에서도 삭제기능을 제공하고, 해당 홈페이지에서도 삭제기능을 제공한다. [그림 2]의 (가)는 다른 소프트웨어를 사용해야하거나, 홈페이지에 찾아갈 필요 없이 삭제기능을 제공하므로 (나)에 비해서 삭제 기능을 수행하기는 용이하다는 것을 알 수 있다.

공인인증서 소프트웨어에 의해서 정상적으로 삭제된 공인인증서가 복구가 가능하여 자신의 공인인증서 파일과 개인키 저장 파일이 어떠한 방법으로든 다른 사람에게 노출되어서는 안 된다. 더욱이 다른 사용자가 정상적인 사용자가 설정한 개인키 암호화 패스워드까지 알아낼 수 있다면 심각한 문제가 아닐 수 없다.

2.2 공인인증서 프로파일

한국에서 사용되는 공인인증서는 X.509 v3 인증서에 대한 프로파일 규격을 따르고 있다. 공인인증서는 크게 기본 필드와 확정필드의 두 부분으로 이루어져 있으며, 본 절에서는 공인인증서 사용자 패스워드(개인키 암호화 패스워드) 검출에 필요한 필드만을 알아본다. 버전 필드는 공인인증서 형식을 구별할 수 있는 기능을 제공하는데 현재 사용하는 공인인증서 규격에는 정수 2를 갖는 v3 인증서만 허용한다[6].

• Version :: INTEGER, Length = 1, Value= 2(0x2)

소유자 공개키 정보 필드는 소유자의 공개키에 대한 알고리즘 및 공개키 정보를 나타낸다. 이 중에서 공개키에 해당하는 필드는 다음과 같다. 141 바이트 중 실질적인 공개키로 사용되는 부분은 128 바이트 이다.

• BIT STRING, Length = 141, Unused bits = 0

2.3 개인키 저장 파일 프로파일

개인키 파일에는 개인키를 암호화할 때 사용되는 비밀키를 생성하는데 쓰이는 솔트(Salt) 값과 반복횟수 값(Iteration Count)이 저장되며, 마지막으로 SEED 블록 암호 알고리즘으로 암호화가 된 개인키 관련 정보들이 들어있다. 솔트 값은 공인인증서를 발급받을 때마다 랜덤하게 생성되는 것으로 비밀키를 생성할 때 랜덤성을 높여 보안성을 강화시키는 역할을 한다. 솔트 값은 21 바이트 ~ 28 바이트 사이의 8 바이트로 이루어져 있다[6].

• OCTET STRING, Length = 8

반복횟수 값은 비밀키 생성을 위한 해쉬함수를 몇 번 수행할 것인가를 나타낸다. 반복횟수 값에 관련된 정보는 31 바이트 ~ 32 바이트 사이의 2 바이트로 이루어져 있다.

• INTEGER, Length = 2, Value = 2048 (0x800)

개인키 정보를 포함하고 있는 필드는 국내 암호화 표준인 SEED 블록 암호 알고리즘으로 암호화되어 있다. SEED 블록 암호 알고리즘에서 사용하는 비밀키는 개인키 암호화 패스워드와 앞에서 설명한 솔트 값, 반복횟수 값 등을 이용하여 만들어지는데, 이는 2.4절에서 자세히 설명하겠다.

• OCTET STRING, Length = 688

2.4 사용자 패스워드를 이용한 개인키의 보호기술

공인인증서에 저장되어 있는 공개키와 쌍을 이루는 사용자의 개인키는 SEED 블록 암호 알고리즘으로 암호화 되어서 개인키 저장 파일에 저장되어 있다. SEED 블록 암호 알고리즘을 이용한 암/복호화에 필요한 비밀키를 생성하기 위해서 사용자가 설정하는 개인키 암호화 패스워드가 사용된다.

한국의 공인인증서 시스템에서는 패스워드 기반의 안전한 개인키 암호화를 위해서 PKCS#5(Public Key Cryptography Standards #5)에서 정의한 PBES1 (Password Based Encryption Scheme 1, 패스워드 기반의 키 암호화 기법)암호화 기법을 이용한다. PBES1에서 정의하고 있지 않은 SEED 블록 암호화 알고리즘을 사

용하기 위하여 암호화 키(K)와 초기 벡터(IV)를 생성하는 방법을 다음과 같은 방법으로 정의된다.

8 바이트의 솔트 값과 반복횟수 값을 선택하고, PBKDF1에 패스워드(P), 솔트 값(S), 반복횟수 값(C)을 적용하여 20 바이트의 추출키(Derived Key)를 생성한다. 한국의 공인인증서 시스템에서는 PBKDF1에서 SHA-1 해쉬함수를 사용한다. 간단히 설명하면 P‖S 값을 SHA-1 해쉬함수에 입력하여 20 바이트의 추출키를 생성하는 작업을 반복횟수 값으로 지정된 횟수만큼 해쉬작업을 수행한다[1]. PBKDF1의 입력값 중 '20' 은 PBKDF1에 사용되는 해쉬함수의 출력 길이를 설정하는 값으로 고정되어 있다.

• DK = PBKDF1(P, S, C, 20)

생성된 추출키(DK)에서 처음 16 바이트를 비밀키(K)로 사용한다.

• K = DK<0 . . 15>

초기벡터의 생성은 아래 2 가지 방식으로 이루어지는데 개인키 파일의 특정 필드 값에 따라서 달라진다. id-seedCBC OBJECT IDENTIFIER가 {1 2 410 200004 1 4}이면, 초기벡터 IV = "30 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35"이며, id-seedCBCWithSHA1 OBJECT IDENTIFIER가 {1 2 410 200004 1 15}이면, DIV = Hash(DK<16 .. 19>)이며 초기벡터 IV = DIV <0 .. 15>이다[1].

그리고 SEED 블록 암호화 알고리즘에 사용되는 패딩은 PBES1에서 정의한 방법을 사용하며 총 16-(‖M‖ mod 16) 바이트의 길이로 구성된다[1]. (‖M‖: 메시지의 길이(바이트))

- ‖M‖ mod 16=15일 때, PS = 01
- ‖M‖ mod 16=14일 때, PS = 02 02
  ....
- ‖M‖ mod 16=0일 때, PS = 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F

Ⅲ. 포렌식 툴을 이용한 공인인증서의 복구

공인인증서를 등록하고 사용하기 위해서는 공인인증서와 개인키 저장 파일, 그리고 공인인증서 관련 파일들을 저장하여야 한다. 현재 주로 사용되고 있는 저장매체는 하드 디스크 드라이브와 USB 드라이브이다. 이곳에 저장된 공인인증서 관련 파일들은 사용자 개인정보이므로 다른 사용자가 쉽게 획득하게 해서는 안 되며, 사용하지 않는 공인인증서 관련 파일들은 사용자가 삭제하여야 한다. 이렇게 공인인증서를 관리하는 상용 소프트웨어들의 기능에는 공인인증서 저장매체의 변경, 인증서 암호 변경, 인증서 보기/검증과 인증서 삭제가 있는데, 조사해본 결과 모든 상용 소프트웨어에서 정상적으로 삭제된 공인인증서도 파일 포렌식 툴을 이용하면 아무런 제약 없이 복구가 가능하다.

〔표 1〕 삭제된 공인인증서와 개인키 저장 파일의 복구

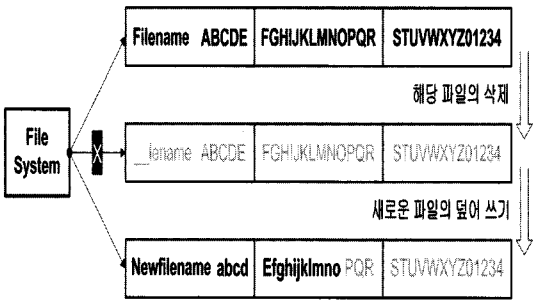
삭제 방식 \ 저장 매체	하드디스크 드라이브	USB 드라이브
공인인증서 소프트웨어를 이용한 삭제	O	O
하드디스크 포맷으로 파일 삭제	O	O
운영체제 상에서의 일반적인 파일 삭제	O	O

더욱이 사용자가 공인인증서를 이용한 서비스를 사용하기 위해서 설치되는 여러 파일들 중에서 공인인증서와 개인키 저장 파일을 제외한 다른 파일들은 새로운 공인인증서를 설치하더라도 변화가 없는 고정된 파일들이다. 그러므로 공격자가 공인인증서와 개인키 저장 파일을 복구할 수 있다면, 사용자의 공인인증서를 도용하여 사용하는데 문제가 없다. 공인인증서와 개인키 저장 파일을 포렌식 툴 중 하나인 복구프로그램을 이용하여 복구한 결과는 다음과 같다. 본 실험에서 사용한 복구프로그램은 FinalData Enterprise v2.01 이다.

위의 〔표 1〕에서 보는 것과 같이 저장매체에 저장된 공인인증서와 개인 키 파일은 공인인증서 소프트웨어를 이용한 삭제 방식, 하드디스크 포맷을 이용한 파일 삭제, 일반적인 삭제 방식을 이용하더라도 각 파일의 완벽한 복구가 가능하다. 그 이유는 저장매체에서 특정 파일을 삭제하면 저장매체에서 파일자체를 삭제하는 것이 아니라, 파일시스템에서 파일과 연결된 링크만을 삭제하기 때문이다. 그리므로 저장매체에는 파일에 관한 정보가 그대로 남아 있어 복구가 가능한 것이다. 공인인증서와 개인키 저장 파일이 완벽히 복구가 가능하다는 것

은 공인인증서 개인키 암호화 패스워드만 알게 되면, 타인의 공인인증서를 이용한 서비스를 아무런 제한 없이 이용할 수 있다는 뜻이다. 공인인증서를 이용하는 서비스는 사용자 개인정보와 관련된 서비스가 많으므로 문제가 된다. 여기서 특히 공인인증서 소프트웨어에서 삭제 메뉴를 통해서 공인인증서를 삭제하였는데도 공인인증서 관련 파일들이 복구가 가능하다는 것은 심각한 문제이다.

그리고 아래의 〔그림 3〕과 같이 저장매체에서 특정 파일을 삭제한 후, 파일이 위치한 부분에 다른 파일이

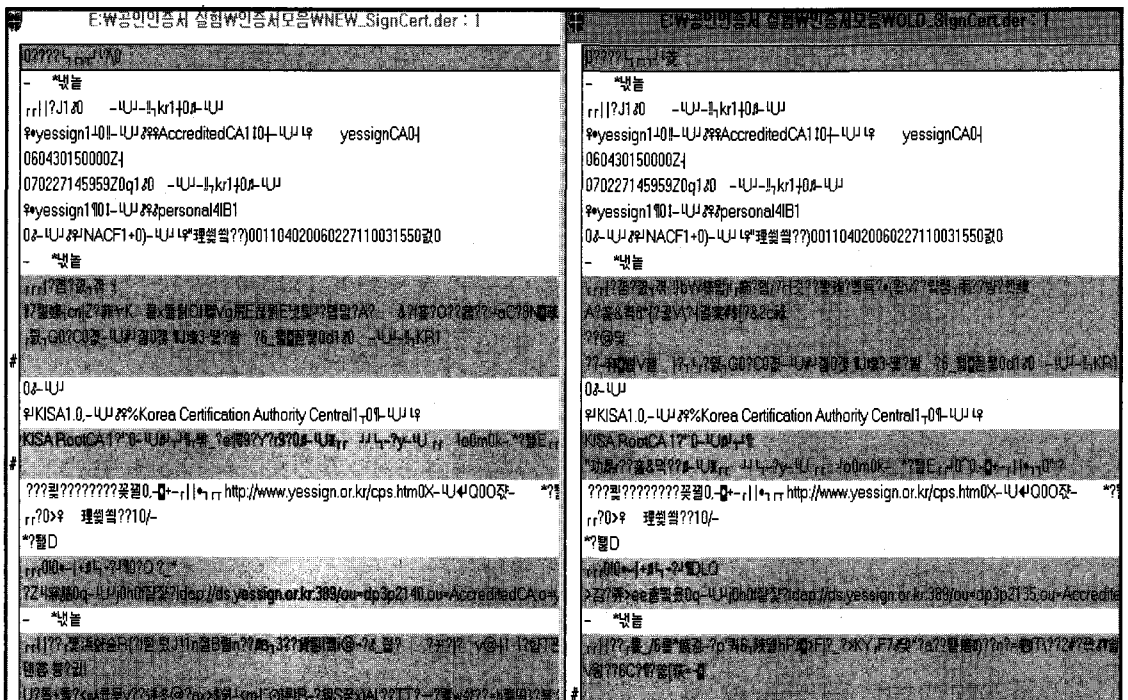


〔그림 3〕 삭제된 파일의 부분정보 복구

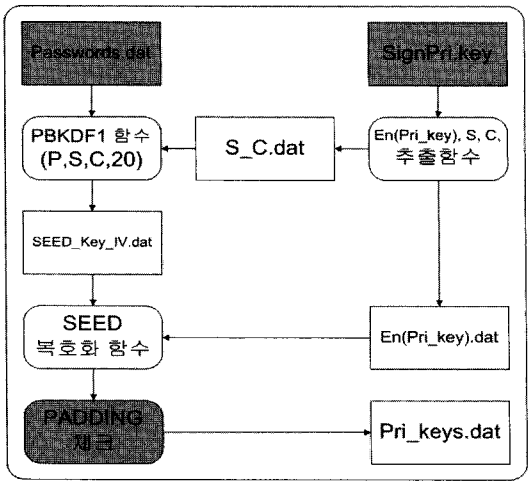
저장되어 덮어쓰기가 된 경우에도 파일의 부분적인 정보만은 여전히 복구가 된다.

〔그림 3〕과 같이 파일 덮어쓰기가 된 경우에는 공인인증서와 개인키 저장 파일의 형태를 같은 파일을 복구해 낼 수는 없지만, 특정한 부분정보들만이 정확히 복구되면 공인인증서와 개인키 저장 파일을 만들어 낼 수 있다. 그것은 관련연구에서 살펴본 것처럼 공인인증서와 개인 키 저장 파일은 일정한 필드로 나누어져 있는데, 필드 값 중에는 모든 인증서에서 같은 값이 사용되는 것이 있기 때문이다. 그러므로 인증서 마다 다른 특정한 필드 값들이 모두 복구된다면, 완벽한 공인인증서와 개인키 저장 파일을 만들어 낼 수 있다. 〔그림 4〕는 다른 2개의 공인인증서의 공통된 부분과 다른 부분을 보여주고 있다.

일부분의 정보만이 남아있을 때는 복구 프로그램을 사용하면 아무런 파일도 복구되지 않는 결과가 나오지만, 저장 매체에는 결국 일부의 정보가 남게 된다. 그때는 비트 단위 검색을 제공하는 WinHex와 같은 포렌식 툴을 이용하여 삭제된 공인인증서와 개인키 저장 파일의 정보를 검색하고 알아낼 수 있다.



〔그림 4〕 공인인증서 파일의 공통된 부분과 다른 부분



〔그림 5〕 오프라인 상의 공인인증서 개인키 암호화 패스워드 검출

Ⅳ. 공인인증서 개인키 암호화 패스워드의 검출

공인인증서 로그인은 특정 홈페이지에 접속할 때 ID-Password 보다 강력한 보안을 제공한다. 공인인증서 사용자의 개인키 암호화 패스워드는 대/소문자를 구분하는 영문자와 숫자를 이용하여 8 자리 이상으로 지정하도록 되어 있다. 실제 동작하는 공인인증서 로그인 소프트웨어에서는 5번 이상 패스워드를 잘못 입력하면 더 이상 해당 공인인증서를 사용할 수 없게 되므로 다른 사용자가 공인인증서 패스워드를 알아내기가 어렵다. 하지만 SEED 블록 암호 알고리즘에 사용되는 패딩의 확인을 통하여 공인인증서의 개인키 암호화 패스워드를 검출할 수 있다.

본 단계에서는 SEED 블록 암호 알고리즘에 사용되는 패딩의 확인을 이용한 개인키 암호화 패스워드의 검출한다. 사용자의 개인키는 SEED 블록 암호 알고리즘을 사용하여 암호화된 후 개인키 파일에 저장되는데, 그때 사용자의 공인인증서 개인키 암호화 패스워드를 이용하여 SEED 블록 암호 알고리즘에 쓰일 비밀키를 생성한다. 그리고 개인키의 전체 입력 사이즈가 SEED 암호 알고리즘의 블록 사이즈의 배수가 되지 않으면 부족한 부분을 채우기 위해서, 공인인증서 소프트웨어에서는 개인키 암호화 패스워드를 이용하여 생성한 비밀키로 SEED 블록 암호 알고리즘을 이용하여 사용자의 개인키를 암호화하는 과정에서 PBES1에서 정의한 패딩을 사용한다. 이것을 이용하면 다른 사용자의 개인키 암호화 패스워드를 알아 낼 수 있는 것이다, 즉 공격자가 임의의 패스워드를 이용하여 생성한 비밀키로 SEED 블록 암호 알고리즘을 이용한 복호화한 후, PBES1에서 정의한 패딩이 존재하는지를 확인으로써 정당한 개인키 암호화 패스워드 인지를 판별할 수 있는 것이다. 〔그림 5〕는 SEED 블록 암호 알고리즘에 사용되는 패딩의 확인을 이용하여, 여러 패스워드 중에서 정당한 개인키 암호화 패스워드를 판별하는 프로그램의 실행과정이다.

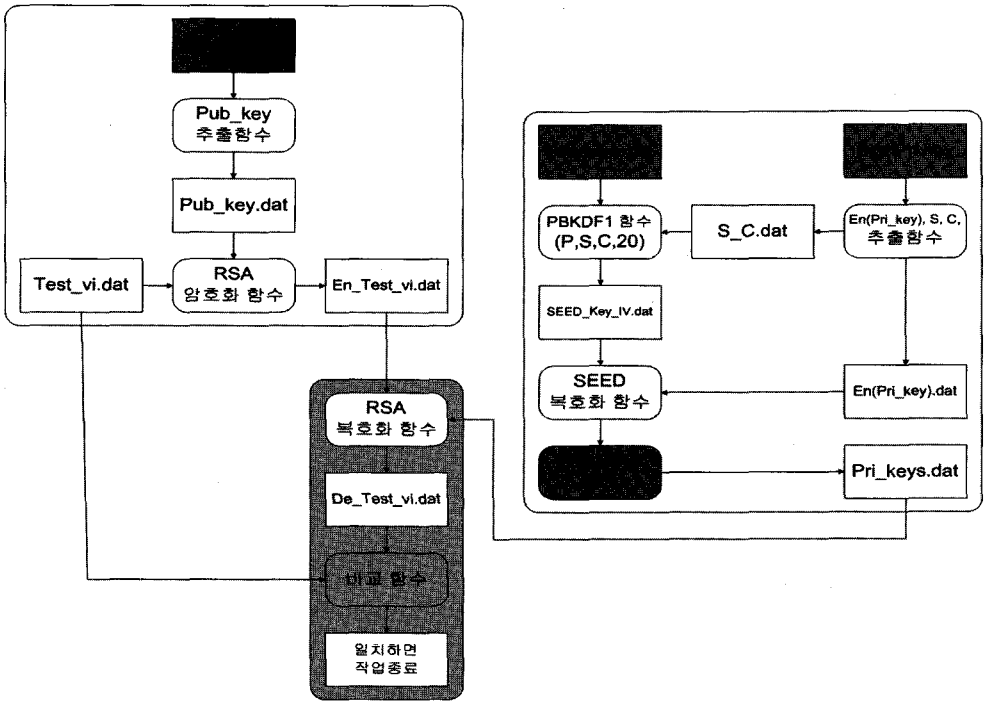
〔그림 5〕에서 Passwords.dat는 공격자가 생성한 개인키 암호화 패스워드 후보들이 저장되어 있는 사전(dictionary)파일이며, SignPri.key는 개인키 저장 파일이다. Passwords.dat에 저장된 패스워드(P)와 개인키 저장 파일에 저장되어있는 8 바이트의 솔트(S) 값, 반복횟수(C) 값, 그리고 원하는 출력 값의 바이트 수를 뜻하는 20을 PBKDF1 함수에 입력값으로 넣어서 SEED 블록 암호 알고리즘에 쓰일 비밀키와 IV를 유도하여 SEED_Key_IV.dat에 저장한다. 그리고 개인키 저장 파일에서 복사하여 En(Pri_key).dat에 저장된 암호화된 개인키 정보를 SEED_Key_IV.dat에 저장된 유도된 비밀키와 초기벡터 값을 사용하여 SEED 블록 암호 알고리즘을 이용하여 복호화를 시킨다. 올바른 패스워드를 사용하여 생성된 SEED 비밀키를 이용하여 복호화된 값들은 PBES1에서 정의한 방법으로 패딩처리 되어 있으므로, 패딩을 체크하여 정확한 패딩이 확인되는 개인키 값을 찾아서 그 해당 개인키 값만을 Pri_keys.dat에 저장한다. 패딩이 확인된 개인키 키 값을 생성할 때 쓰인 패스워드가 정당한 공인인증서 개인키 암호화 패스워드라고 할 수 있다.

하지만 패딩의 확인을 통한 방식은 복호화된 개인키의 사용 가능여부가 아닌 오직 복호화된 값의 패딩 존재여부만을 확인하는 것이기 때문에, 잘못된 개인키 암호화 패스워드로 생성한 비밀키로 암호화된 개인키 정보를 복호화 한 값에 우연히 정당한 패딩 값 형태의 정보가 존재할 수 있다. 그래서 복호화된 개인키 값을 검증하는 과정이 필요한 것이다.

Ⅴ. 공인인증서 개인키 암호화 패스워드의 검증

공인인증서 개인키 암호화 패스워드의 검증은 공인인증서에 저장되어 있는 사용자의 공개키와 개인키 파일에 저장되어있는 사용자 개인키의 연관성 확인하는 과정으로 이 단계를 통하여 정당한 개인키 암호화 패스워드인지를 정확히 판별해 낼 수 있다.

본 단계는 이렇게 정당한 개인키 암호화 패스워드를



〔그림 6〕 오프라인 상의 공인인증서 개인키 암호화 패스워드 검출 및 검증

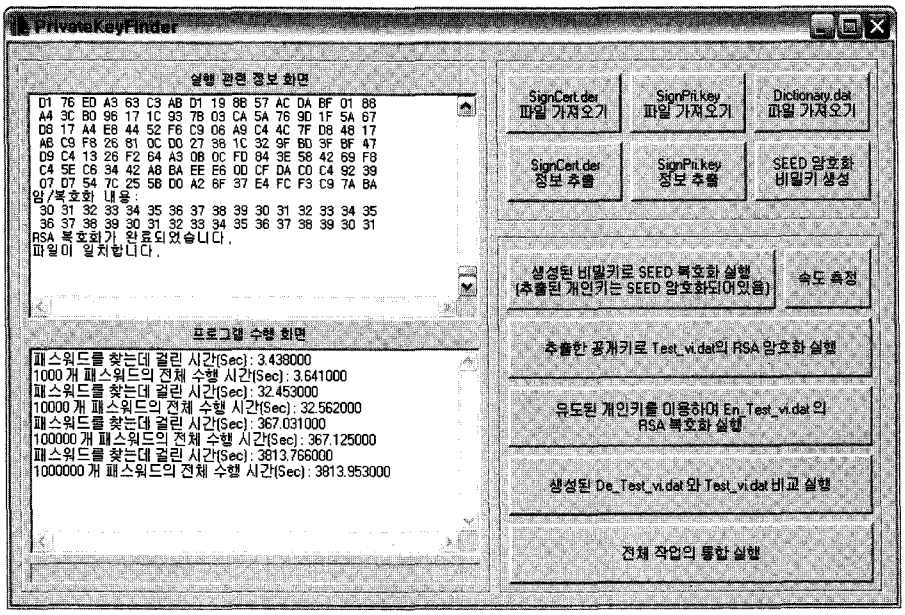
검증하기 위한 것으로 패딩 체크로 얻어진 개인키 값이 정확한지를 확인하는 과정이다. 공인인증서에서 가져온 사용자 공개키를 이용하여 테스트 벡터(Test_vi.dat)를 RSA 방식으로 암호화한 후, Pri_keys.dat에 저장된 개인키로 다시 RSA 복호화를 하여 나온 결과 값과 테스트 벡터를 비교하는 방식이다. 〔그림 6〕은 패딩체크를 통한 개인키 암호화 패스워드의 검출 후 생성된 개인키가 사용가능하지를 체크하여 정확한 개인키 암호화 패스워드를 알아내는 검증 프로그램의 실행과정을 보여준다.

〔그림 6〕에서 SignCert.der는 사용자의 공인인증서이며, SignCert.der에서 저장되어있는 공개키를 이용하여 테스트 벡터(Test_vi.dat)를 암호화한 값은 En_Test_vi.dat에 저장된다. 그리고 De_Test_vi.dat은 En_Test_vi.dat에 저장된 값을 Pri_keys.dat에 저장된 개인키로 다시 RSA 복호화를 하여 나온 결과 값을 저장한다. 비교함수에서는 Test_vi.dat에 De_Test_vi.dat에 저장된 값이 일치하는지 비교하는 함수이다. 비교함수를 통해서 Test_vi.dat와 De_Test_vi.dat에 저장된 값들이 일치하다는 것이 확인되면, 해당하는 개인키 암호화 패스워드가 사용자가 설정한 패스워드임이 검증된다.

패스워드 검출 및 검증하는 프로그램을 정리하면, 본 프로그램은 SEED 블록 암호 알고리즘에 사용되는 패딩의 확인을 통하여 개인키 암호화 패스워드를 검출한 후, 보다 정확한 검증을 위해서 공인인증서에 들어있는 사용자 공개키와 개인키 저장 파일에 들어있는 사용자 개인키의 관련성을 이용하여 사용자의 공인인증서 개인키 암호화 패스워드를 확인하는 방식으로 이루어져있다. 실험에 사용된 패스워드들 중 사용자의 패스워드와 일치하는 것을 검출하기 위해서 〔그림 5〕와 같이 PBDKF1 함수와 SEED 복호화 함수를 이용하여 생성된 후 패딩 체크까지 끝난 개인키 후보들을 생성하고 그에 해당하는 패스워드를 알아낸다. 그리고 더 정확한 검증을 위해서 〔그림 6〕과 같이 테스트 벡터를 사용자 공개키로 RSA 암호화한 후 그 값을 Pri_keys.dat에 있는 〔그림 5〕에서 검출된 정확한 개인키를 이용하여 RSA 복호화한 후, 그 결과 값과 처음 테스트 벡터와 일치하는 지를 확인한다.

Ⅵ. 공인인증서 개인키 암호화 패스워드 검출 프로그램 성능평가

공인인증서와 개인키 저장 파일을 이용하여 공인인



〔그림 7〕 오프라인 상의 공인인증서 개인키 암호화 패스워드 검출 및 검증 프로그램

증서 개인키 암호화 패스워드를 검출하는 프로그램은 오프라인 상에서 횟수의 제한 없이 사용 가능하다. 〔그림 7〕은 공인인증서 개인키 암호화 패스워드 검출 프로그램의 실행화면이다.

패스워드 검출 속도는 프로그램을 이용하여 해당하는 패스워드를 검출해낼 때까지의 속도이며, 패스워드 검출/검증 속도는 패스워드 검출뿐만 아니라 공인인증서의 공개키와 검출한 개인키로 검증하는데 걸리는 시간이다. 검출/검증 프로그램에 사용되는 패스워드 파일에 저장된 패스워드들 중 마지막 패스워드를 실제 패스워드로 설정하였다. 그러므로 실험결과로 나온 프로그램의 속도가 각각 1,000개/10,000개/100,000개/1,000,000개의 패스워드를 검사하는데 걸리는 시간이라고 할 수 있다. 다음의 〔표 2〕는 프로그램의 속도를 보여주고 있다. 해당 프로그램의 속도는 실행되는 컴퓨터의 기본적인 속도 및 실행환경조건(타 프로그램의 실행여부, CPU 점유율 등)에 따라서 유동적이지만 대략적인 패스워드 검출/검증 시간을 알아보는 것에는 도움이 된다.

〔표 2〕 패스워드 종류에 따른 프로그램 속도

기준 / 개수	패스워드 검출 속도	패스워드 검출/검증 속도
1,000 개	3.438 (Sec)	3.641 (Sec)
10,000 개	32.453 (Sec)	32.562 (Sec)
100,000 개	367.031 (Sec)	367.125 (Sec)
1,000,000 개	3,813.766 (Sec)	3,813.953 (Sec)

본 프로그램은 〔그림 5〕와 같이 하나의 패스워드를 이용하여 SEED 복호화 알고리즘 용 비밀키를 생성한 후, 해당 비밀키를 이용하여 암호화되어 있는 사용자의 개인키를 복호화한 후 Padding을 체크하여 KISA(Korea Information Security Agency)에서 지정한 표준과 일치하는지를 검사하는 과정을 검출이라고 정하였으며, 〔그림 6〕과 같이 공인인증서에서 저장된 사용자 공개키와 KISA에서 지정한 표준과 일치하는 Padding이 있는 해당 개인키만을 이용하여 개인키를 진위여부를 확인하는 과정을 검증이라고 하였다. 그리고 1,000개 패스워드 중 마지막 패스워드가 실제 패스워드인 것을 가만할 때, 1,000개의 패스워들 검사할 때는 〔그림 5〕의 과정은 999번, 〔그림 6〕의 과정을 1번 수행한다. 10,000개의 패스워드를 검사할 때는 〔그림 5〕의 과정은 9,999번, 〔그림 6〕의 과정을 1번 수행한다.

현재 공인인증서 시스템에서 지정하는 비밀번호는 8자리 이상이다. 그리고 선택 가능한 문자는 대/소문자를 구분하는 영문자 52 가지와 숫자 10 가지, 그리고 빈칸과 특수문자 32 가지로 총 94개이다. 그러므로 비밀번호가 8 자리일 때 설정가능 사용한 비밀번호 경우의 수는 (94)8 가지 : 6,095,689,385,410,816 가지이다. 이것을 키 보안강도와 비교하면 50 비트 키의 길이와 보안

강도가 비슷하다. 사용자가 비밀번호의 길이를 10 자리로 설정한다고 하더라도 62 비트의 키 길이 정도의 보안성을 가진다. 하지만 현재 실질적으로 사용되는 있는 패스워드를 분석하면, 실질적인 패스워드 보안성은 50~62 비트 보안성보다 낮다는 것을 알 수 있다. 현재 사용자들이 사용하고 있는 패스워드의 유형과 그 경우의 수를 보면 다음과 같다. 패스워드 검출 시간은 〔표 2〕의 결과를 종합하여, 패스워드 1,000,000개 검출하는데 4,000 초가 걸린다고 가정한다. 〔표 3〕에서는 공인인증서 패스워드 유형에 따른 패스워드 최대 검출 시간을 보여주고 있다. 미국 컴퓨터보안연구소인 SANS (SysAdmin, Audit, Network, Security)에서 안전한 패스워드를 생성하기 위한 패스워드 정책과 마이크로소프트의 패스워드 설정 원칙 및 미국 NIST 전자인증 가이드라인 등을 참조하여 공인인증서에서 자주 사용되고 약한 패스워드라고 할 수 있는 패스워드의 유형별 패스워드의 최대 검출시간에 대한 실험결과를 〔표 3〕에서 보여주고 있다.[17,18,19]

본 프로그램의 실행된 PC의 사양은 CPU Pentium 4 2.4와 RAM 1 GB 이다. 다음과 같은 실험환경에서 공격자는 사용자가 선택할 수 있는 가장 약한 패스워드 유형인 8 자리_영소문자1_숫자7 로 이루어진 사용자 개인키 암호화 패스워드는 5시간 안에 260,000,000 가지 모든 패스워드 종류를 검사해 볼 수 있었다. 그리고 8 자리_영소문자4_숫자4 인 설정된 경우에는 모든 경우의 수는 4,569,760,000 가지이므로 서 개인키 암호화 패스워드를 검출하는데 5077시간, 즉 212일밖에 걸리지 않는다. 영문자로만 구성하는 형태인 8 자리_영문자4_숫자4 로 패스워드가 구성되었을 때는 총 경우의 수는 208,827,064,576 가지로 본 PC 환경에서 모든 경우의 수를 검사하는데 26년이라는 시간이 소요되었다. 하지만 개인키 암호화 패스워드를 검출하는 본 실험이 PC 1대에서 진행되었고, PC의 사양이 높아지면 검출 속도도 빨라진다는 것을 감안하면 안전하다고 할 수 없다. 그리고 공인인증서가 폐기되더라도 사용자가 공인인증서 패스워드를 변경하지 않고 그대로 유지하는 경우가 많다. 그래서 한번 노출된 공인인증서 패스워드는 새롭게 발행한 공인인증서에도 영향을 미치므로 안전하게 보호되어야 한다.

〔표 3〕 패스워드 유형에 따른 패스워드 최대 검출 시간

패스워드유형	모든 경우의 수	패스워드 최대 검출 시간
영소문자1_숫자7 Ex) s1234567	$(26)^1 * (10)^7$ = 260,000,000	289분/ 4시간 49분
영소문자4_숫자4 Ex) choi0420	$(26)^4 * (10)^4$ = 4,569,760,000	5,077시간 / 212일
영문자4_숫자4 Ex) GooD1212	$(52)^4 * (10)^4$ = 73,116,160,000	81,240시간 / 3,385일 / 9년
영소문자8 Ex) choiyoun	$(26)^8$ = 208,827,064,576	232,030 시간 / 9,668일 / 26년
소문자5_숫자4 Ex) cyber0815	$(26)^5 * (10)^4$ = 118,813,760,000	132,015시간 / 5,501일 / 15년
소문자6_숫자4 Ex) system1004	$(26)^5 * (10)^4$ = 3,089,157,760,000	3,432,397시간 / 143,017일 / 392년

Ⅶ. 공인인증서 복구 및 개인키 암호화 패스워드 검출 방지 방안

본 장에서는 3장에서 분석한 현재 공인인증서 시스템의 문제점을 보안하기 위한 방법을 제안한다.

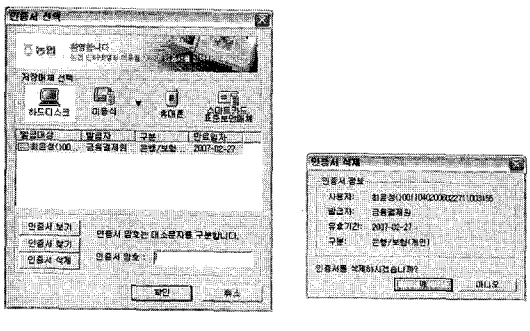
7.1 공인인증서의 복구 방지

현재 공인인증서 관리 소프트웨어는 소프트웨어들 종류와 이용하는 방식에 따라서 삭제 기능을 이용할 수 있는 접근성이 조금씩 차이가 있다. 사용자가 현재 사용하고 있는 컴퓨터나 USB 저장매체에 저장된 공인인증서를 더 이상 사용하지 않을 경우에 공인인증서의 완전한 삭제는 필수적임에도, 공인인증서의 삭제기능을 이용 하는데 불편할 뿐만 아니라, 삭제기능을 이용하더라도 완전한 삭제가 이루어지지 않는다. 그러므로 사용자가 공인인증서의 삭제기능을 이용하기가 용이하고, 삭제기능을 실행한 뒤에 복구가 불가능하게 완전한 삭제가 이루어져야 한다. 공인인증서 관리 소프트웨어와 유사한 PGP 프로그램에서는 Wiping 기술을 지원한다. PGP 프로그램을 설치하면 선택한 파일을 Wiping 기술을 이용하여 완전히 삭제할 수 있게 지원하는 것이다. PGP 프로그램 설치 후, 특정 파일이나 폴더를 선택한 후 마우스의 오른쪽 버튼을 누르면 [Delete with Wiping] 이라는 메뉴가 나타난다. 이와 같이 공인인증서 관리 프로그램에서도 Wiping 기술을 이용한 완전한 파일 삭제를 지원할 필요가 있다. 그러므로 〔그림 8〕에서 보이는 메뉴 [인증서 삭제]가 제공하는 공인인증서의 삭제기능에 Wiping 기술이 기본적으로 제공될 필요가 있다.

그리고 공인인증서 삭제기능을 수행한 후, 공인인증서의 복구를 방지하기 위해서는 상용 공인인증서 소프트웨어 삭제기능에 데이터 완전삭제(wiping) 기술이 적용되어야 한다. 데이터 완전삭제 기술은 저장매체에 한 번 삭제된 중요 데이터가 복구되어 악용되는 것을 막기 위해서 저장매체에서 데이터를 완전히 삭제하여 준다. 아래에서 설명하는 데이터 완전삭제 기술은 하드디스크를 기준으로 개발된 것이지만 다른 저장매체에서도 적용할 수 있다. 다양한 데이터 완전삭제 기술들이 있지만, 높은 안전성을 요구하는 데이터를 완전히 삭제하는 알고리즘으로 7번 덮어쓰기(7-Passes)가 있으며 2 가지 방식이 있다[16].

첫 번째로 미 국방성의 데이터 완전삭제 알고리즘이 있다. 이 방식은 처음에 데이터 저장 장소에 '01010101'을 덮어 쓰고, 다음에 '10101010'을 덮어 쓰고, 앞의 두 단계를 3번 반복한다. 그리고 마지막으로 랜덤한 데이터를 덮어 쓰는 방식이다. 두 번째 방식으로는 Bruce Schneier의 알고리즘이 있다. 이 방식은 Bruce Schneier가 자신의 저서인 Applied Cryptography에서 소개한 데이터 완전삭제 알고리즘이다. 먼저 HDD내 모든 데이터 저장 장소를 '11111111'로, 다음으로 '00000000'로 덮어쓴다. 그리고 암호학적으로 안전한 의사 난수 발생기에서 발생한 난수를 5 번 덮어쓴다.

현재 공인인증서 소프트웨어에서 공인인증서 및 개인키 저장 파일을 삭제하고자 할 때는 [그림 8]과 같은 방식과 같이 공인인증서 소프트웨어에서 인증서 삭제 메뉴를 선택하면 된다. 하지만 현재 사용되는 인증서 삭제 프로그램으로 삭제된 공인인증서와 개인키 저장 파일은 포렌식 툴에 의해서 복구가 가능하므로, 인증서 삭제 알고리즘에 데이터 완전삭제 기술을 적용하여야 한다. 공인인증서 및 개인 키 파일의 크기는 각각 2KB, 1KB 로 작은 편이지만 매우 중요한 정보를 지니고 있으므로, 미 국방성의 데이터 완전삭제 알고리즘이나 Bruce Schneier의 알고리즘을 이용한 데이터 완전삭제 기술을 통하여 공인인증서와 개인키 저장 파일을 삭제하여야 한다. 미국 국방성의 데이터 완전삭제 알고리즘과 Bruce Schneier의 알고리즘은 속도는 느리지만, 완벽한 삭제가 가능하므로 파일의 크기가 작고 중요한 정보를 지닌 파일의 완전삭제에 적합하다.



[그림 8] 공인인증서 소프트웨어에서 공인인증서 및 개인

7.2 공인인증서 개인키 암호화 패스워드 검출 방지

현재 발급된 공인인증서의 유효기간은 발급일로부터 1년이다. 공격자가 정당한 사용자의 공인인증서를 취득하고 개인키 암호화 패스워드를 알아내더라도 공인인증서의 유효기간이 지나면 효용가치가 없어진다. 그러므로 약한 공인인증서 개인키 암호화 패스워드가 생성되는 것을 막는 규칙을 지정하여 공인인증서 유효기간 내에 공인인증서 개인키 암호화 패스워드를 검출하기 어렵게 하여야 한다.

현재 공인인증서 개인키 암호화 패스워드 생성 규칙은 영문자 하나를 포함한 8 자리이상의 패스워드이다. 사용자가 개인키 암호화 패스워드를 설정할 때는 사용가능한 문자의 수는 94 가지이지만, 사용자가 주로 사용하는 문자는 영소문자와 숫자이다. 그러므로 사용자가 개인키 암호화 패스워드를 설정할 때, 영대문자 혹은 특수문자가 필수적으로 포함되어야 한다는 것과 같은 패스워드 정책을 지정하거나, 공인인증서 관리 시스템에서 사용자에게 패스워드를 지정해주는 방법이 있다.

패스워드 정책을 이용한 개인키 암호화 패스워드 강화방식은 다음과 같다. 먼저 사용자가 개인키 암호화 패스워드를 설정할 때 정해진 패스워드 정책을 사용자에게 보여준다. 그리고 정책에 만족하지 않는 패스워드에 대해서 패스워드의 보안성이 취약하다는 경고 메시지를 보여줌으로써 재설정을 권고하거나, 정책에 만족하지 않는 패스워드는 처음부터 설정되지 않도록 하는 것이다. 패스워드 설정 원칙은 마이크로소프트의 패스워드 설정 원칙 및 미국 NIST 전자인증 가이드라인 등을 참조할 수 있다[17,18]. 그리고 미국 컴퓨터보안연구소인 SANS 에서 안전한 패스워드를 생성하기 위한 패스워드 정책을 제시하였는데, 이 정책은 안전한 개인키 암호화 패스워드를 생성에 적용해도 무방하다. SANS의 패스워드 정책은 부록으로 첨부한다[19].

〔표 4〕 패스워드 부분정보를 이용한 안전한 패스워드의 생성

사용자가 설정하는 패스워드 길이	패스워드 부분정보 형태	추가되는 패스워드	모든 경우의 수	패스워드 검출 시간
8 자리	aa9	?????	$(26)^2 \times (10)^1 \times (95)^5$ = 52,307,591,375,000	209,230,365,500 초 3,487,172,758 분 / 58,119,546 시간 2,421,648 일 / 6,635 년
	&&&	?????	$(32)^3 \times (95)^5$ = 253,552,537,600,000	1,014,210,150,400 초 16,903,502,507 분 / 281,725,042 시간 11,738,543 일 / 32,160 년
9 자리	aa99	?????	$(26)^2 \times (10)^2 \times (95)^5$ = 523,075,913,750,000	10,142,101,504,000 초 16,903,502,507 분 / 2,817,250,420 시간 117,385,430 일 / 321,600 년
10 자리	999999	???	$(10)^6 \times (95)^4$ = 81,450,625,000,000	325,802,500,000 초 5,430,041,666 분 / 90,500,694 시간 4,525,035 일 / 12,397 년

또 하나의 개인키 암호화 패스워드 강화방식은 공인인증서 관리 시스템에서 패스워드의 전체 혹은 부분을 생성해주는 것이다. 사용자의 개인키 암호화 패스워드 전체를 공인인증서 관리 시스템에서 생성하여 주면 가장 안전하다는 장점이 있지만, 설정된 패스워드를 사용자가 기억하기가 어렵다는 단점이 있다. 그래서 다른 방법으로 사용자에게 원하는 패스워드 길이와 패스워드 부분정보를 입력받아 시스템에서 직접 강력한 개인키 암호화 패스워드를 생성하여 주는 방식이 있다.

즉 사용자가 강력한 개인키 암호화 패스워드를 생성할 수 있게 시스템에서는 사용자에게 원하는 패스워드 길이와 패스워드의 부분정보만을 입력받아서, 입력받은 패스워드 부분정보에 적합한 랜덤한 패스워드를 추가시킴으로서 사용자가 원하는 길이의 강력한 개인키 암호화 패스워드를 생성하여 주는 것이다. 다음 〔표 4〕는 사용자에게 입력받은 패스워드 부분정보와 원하는 패스워드 길이를 이용하여 안전한 패스워드들을 생성하는 예시들이다. 'a' 는 영소문자, '9' 는 숫자, '&' 는 특수문자는 사용자에게 입력받은 패스워드 부분정보의 형태를 나타내며, '?' 는 프로그램에 의해서 랜덤하게 생성되는 패스워드 부분을 나타내고 있다.

Ⅷ. 결 론

본 논문에서는 현재 공인인증서 시스템에서 정상적으로 삭제된 공인인증서와 개인 키 정보가 저장되어 있는 개인키 저장 파일이 상용 포렌식 툴을 이용하면 아무런 제약 없이 복구되는 것을 살펴보았으며, 복구된 공인인증서와 개인 키 파일을 이용하여 오프라인 상에서 사용자의 공인인증서 개인키 암호화 패스워드를 찾아내고 그것을 검출해낼 수 있는 방법을 설명하였다. 개인키의 암호화하여 저장할 때 사용자 패스워드를 이용하지 않는 새로운 방안이 없는 이상은 공인인증서의 개인키 암호화 패스워드 검출은 피해갈 수 없는 문제이다. 그러므로 공인인증서 소프트웨어는 삭제기능에 데이터 완전삭제(wiping) 기술을 적용하여야 한다. 그리고 검출에 약한 개인키 암호화 패스워드가 사용되지 않도록 하는 강력한 패스워드 정책의 수립이 필요하다.

참고문헌

〔1〕 정보보호진흥원 암호인증기술팀, SEED 알고리즘을 이용한 개인키 암호화 기술규격 〔v1.00〕, 정보보호진흥원, 2004.

〔2〕 정보보호진흥원 암호인증기술팀, 전자서명 인증서 프로파일 기술규격 [V1.10], 정보보호진흥원, 2004.

〔3〕 R. Hunt. PKI and digital certification Infrastructure, Ninth IEEE International Conference on Networks (ICON'01), October, 2001.

〔4〕 최희봉, 오수현, 홍순좌, 원동호, PKI 연동 키복구 암호 시스템 설계에 관한 연구, 정보보호학회논문지 12(1), pp. 11-19, 2002.

〔5〕 T. Auro and C. Ellison. Privacy and Accountability in certificate Systems, Helsinki University of Technology, 2000.

〔6〕 정보보호진흥원 암호인증기술팀, 공인인증서 표시를 위한 기술규격[V1.00], 정보보호진흥원, 2002.
〔7〕 J. S. Park and R. Sandhu. Binding identities and attributes using digitally signed certificates, Annual Computer Security Applications Conference 2000, USA , 2000.
〔8〕 이래, 이동훈, 코드 서명 기술의 국내 PKI 적용 방안 비교 연구, 정보보호학회논문지 14(3), pp. 13-27, 2004.
〔9〕 추경균, 김종배, 류성열, 정부의 행정전자서명인증체계(GPKI) 활성화 및 발전방안, 정보보호학회논문지 14(2), pp. 85-100, 2004.
〔10〕 염흥열, 정보보호 법제도 및 기술 표준화 : PKI 표준화 동향과 PKI 영역간 상호 연동 방법, 정보보호학회지, 12(4), pp. 23-46, 2002.
〔11〕 Sharon Boyen, Tim Howes and Patrick Richard. Internet X.509 Public Key Certificate Operational Protocols-LDAPv2. RFC2559, IETF Network Working Group, April 1999.
〔12〕 William Burr, Donna Dodson, Noel Nazario, W. Timothy Polk. MISPC Minimum Interoperability Specification for PKI Components, Version 1 Computer Science Resource Center, NIST September, 1997.
〔13〕 심주결, 박택진, 이철원, 원동호, 국내 PKI 시스템 평가 기준 제안, 정보보호학회논문지, 12(3), pp. 45-61, 2002.
〔14〕 정보보호진흥원 암호인증기술팀, 전자서명 인증서 효력정지 및 폐지목록 프로파일 기술규격 [V1.10], 정보보호진흥원, 2004.
〔15〕 김영백, 이석래, 이재일, 고승철, 전자서명 키관리 시스템에 대한 고찰, 정보보호학회지, 10(4), pp. 1-8, 2000.
〔16〕 Intrusion Detection, Diagnosis, and Recovery with Self-Securing Storage. John D. Strunk, Garth R. Goodson, Adam G. Pennington, Craig A.N. Soules, Gregory R. Ganger. CMU SCS Technical Report CMU-CS-02-140, May 2002.
〔17〕 마이크로소프트의 패스워드 설정 원칙, http://www.microsoft.com/athome/security/privacy/password.mspx
〔18〕 미국 NIST 전자인증 가이드라인 표준, NIST Special Publication 800-63-Appendix A, http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf
〔19〕 The SANS Institute, Password Protection Policy Standards Organization, http://www.sans.org/resources/policies/Password_Policy.pdf

부록 1. SANS의 패스워드 정책

SANS Institute의 보안 정책 프로젝트에서는 사용자, 관리자 및 개발자를 위한 강력한 패스워드를 만드는 데 도움을 주는 무료 안내서를 제공하고 있으며 주요 내용은 다음과 같다.

1. 일반적인 정책

- 모든 시스템 레벨의 패스워드들(예. root, enable, NT admin, application administration accounts, 등) 은 적어도 분기당 한 번씩 변경되어야한다.
- 모든 생산 시스템 레벨의 패스워드들은 반드시 정보보호 관리자적인 포괄적인 패스워드 관리 데이터베이스의 한 부분이어야 한다.
- 모든 사용자 레벨의 패스워드들(예. E-mail, web, desktop computer, 등)은 적어도 6개월에 한번 씩 변경되어야한다. 권장하는 변경 기간은 4개월에 한 번씩이다.
- 특정 그룹의 멤버로써나 sudo와 같은 프로그램을 이용하여 시스템 레벨의 권한이 가지게 된 사용자 계정은 그 사용자가 가진 다른 계정과는 다른 유일한 패스워드를 사용하여야 한다.
- 패스워드는 E-mail 메시지나 다른 전자통신에 첨부되어 전송되어서는 안 된다.
- SNMP를 사용하는 곳에서는 기본적으로 설정되어 있는 Public, Private, System 과는 다른 정의된 커뮤니티 스트링을 정의하여야 하고 로그인에 영향을 미치는 패스워드와는 다른 패스워드를 사용하여야 한다.
- 모든 사용자 레벨 및 시스템 레벨 패스워드는 반드시 아래에서 설명하는 가이드라인에 따라 생성한 패스워드를 사용하여야 한다.

2. 약한 패스워드 조건

- 8 자리 이하의 패스워드
- 사전에 있는 단어로 이루어진 패스워드(영어사전 or 외국어사전)
- 다음과 같이 주로 사용되는 단어로 이루어진 패스워드
  - 가족들, 애완동물, 친구들, 기업의 이름으로 이루어진 패스워드
  - 컴퓨터 용어나 명령어, 사이트, 회사, 하드웨어, 소프트웨어의 이름으로 이루어진 패스워드
  - 생일이나 주소나 휴대전화번호 등과 같은 개인정보로 이루어진 패스워드
  - 패턴이 존재하는 패스워드 (Ex. aaabbb, qwerty, zyxwvuts, 123321 등)
  - 영어 단어의 스펠링을 거꾸로 적은 패스워드
  - 하나의 숫자가 제일 앞이나 제일 뒤에 오는 패스워드 (Ex. security1, 1security)

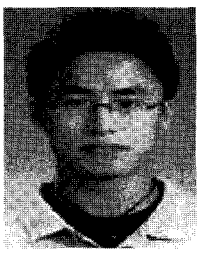
3. 강한 패스워드 조건

- 대문자와 소문자가 모두 포함된 패스워드
- 숫자와 특수문자가 포함된 패스워드
- 적어도 8 자리 이상의 길이로 이루어진 패스워드
- 특정 언어의 단어, 속어, 방언, 은어로 이루어진 패스워드
- 개인정보를 기초로 단어와 가족의 이름이 포함되지 않은 패스워드
- 노래 제목이나 명언, 혹은 다른 문구를 기초로 한 패스워드
  - 예를 들어 "This May Be One Way To Remember"를 기초로 하여한 패스워드의 예로는 "TmB1w2R" 이나 "Tmb1w>r~" 와 같은 형태가 있다.

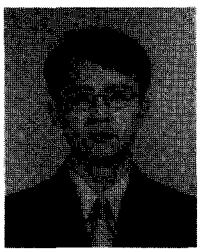
4. 패스워드의 보호를 위한 금지 사항

- 누구에게도 전화로 패스워드를 누설해서 안 된다.
- E-mail 내용에 패스워드를 적어서는 안 된다.
- 상사에게 패스워드를 누설하면 안 된다.
- 다른 사람 앞에서 패스워드에 관해서 이야기하면 안 된다.
- 패스워드 형태에 관한 정보 및 힌트를 주어서는 안 된다.
- 질문서나 보안 양식에 패스워드를 적으면 안 된다.
- 가족과 패스워드를 공유해서는 안 된다.
- 휴가기간 동안 직장 동료와 비밀번호를 공유하여서는 안 된다.

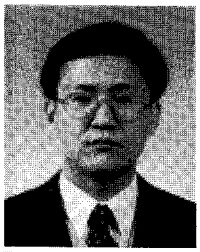
〈著者紹介〉



최 윤 성 (Younsung Choi) 학생회원
2006년 2월 : 성균관대학교 정보통신공학부(공학사)
2006년-현재 성균관대학교 일반대학원 전자전기컴퓨터공학과 석사과정 재학 중
<관심분야> 디지털 포렌식, 정보보호 응용, PKI, 보안성 평가



이 영 교 (Younggyo Lee) 학생회원
1986년 2월 : 한양대학교 전자공학과 (공학사)
1991년 6월 : 한양대학교 대학원 전자공학과 (공학석사)
2002년 3월-현재 : 성균관대학교 대학원 전기전자 및 컴퓨터공학부 박사수료
1993년 3월-1998년 9월 : 대우통신 종합연구소 선임연구원
1999년 2월-2001년 6월 : LG 전자/정보통신 중앙연구소 선임연구원
2002년 3월-현재 : 인하공업전문대학 정보통신과 강사
2004년 9월-현재 : 아주대학교 정보통신대학원 강사
<관심분야> 암호이론, 정보통신 보안, 네트워크 이론



이 윤 호 (Yunho Lee) 학생회원

1991년 2월:성균관대학교 정보공학과(공학사)
1993년 2월:성균관대학교 대학원 정보공학과(공학석사)
1993년 3월-2000년 4월:한국통신 연구개발본부 전임연구원
2000년 5월-2005년 1월:KBS인터넷㈜ 기술지원팀장
2005년 3월-현재:성균관대학교 컴퓨터공학과 박사과정 재학 중
2006년 6월-현재:㈜애니온소프트 기술이사
<관심분야> 암호이론, 정보보호 응용, 전자투표, 워터마킹

박 상 준 (Sangjoon Park) 종신회원

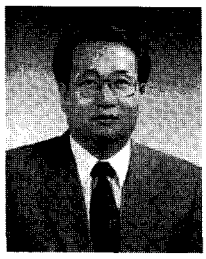
1986년 2월: 한양대학교 수학과 석사
1999년 2월: 성균관대학교 정보공학과 박사 (암호전공)
1986년 1월~1999년 12월: 한국전자통신연구소 부호기술부 선임연구원
2000년 1월~2000년 10월: 국가보안기술연구소 책임연구원
2000년 11월~2005년 5월: (주)비씨큐어 부사장
2005년 7월~현재: 성균관대학교 정보보호기술연구소 연구교수
<관심분야> 암호 알고리즘, 키 분배, 인증 및 서명, 암호분석

양 형 규 (Hyungkyu Yang) 종신회원

1983년~1985년: 성균관대학교 전자공학과(학사,석사)
1995년: 성균관대학교 정보공학과(박사)
1984년~1991년: 삼성전자 컴퓨터부문 선임연구원
1995년~현재: 강남대학교 컴퓨터미디어공학부 부교수
<관심분야> 암호 프로토콜, 네트워크 보안

김 승 주 (Seungjoo Kim) 종신회원

1994년 2월~1999년 2월: 성균관대학교 정보공학과 (학사, 석사, 박사)
1998년 12월~2004년 2월: 한국정보보호진흥원(KISA) 팀장
2004년 3월~현재: 성균관대학교 정보통신공학부 교수
2001년 1월~현재: 한국정보보호학회, 한국인터넷정보학회, 한국정보과학회, 한국정보처리학회 논문지 및 학회지 편집위원
2002년 4월~현재: 한국정보통신기술협회(TTA) IT 국제표준화 전문가
2005년 6월~현재: 교육인적자원부 유해정보차단 자문위원
<관심분야> 암호이론, 정보보호표준, 정보보호제품 및 스마트카드 보안성 평가, PET



원 동 호 (Dongho Won) 종신회원

1976년~1988년: 성균관대학교 전자공학과(학사, 석사, 박사)
1978년~1980년: 한국전자통신연구원 전임연구원
1985년~1986년: 일본 동경공업대 객원연구원
1988년~2003년: 성균관대학교 교학처장, 전지전자 및 컴퓨터공학부장, 정보통신대학원장, 정보통신기술연구소장, 연구처장.
1996년~1998년: 국무총리실 정보화추진위원회 자문위원
2002년~2003년: 한국정보보호학회장
현재: 성균관대학교 정보통신공학부 교수, 한국정보보호학회 명예회장, 정보통신부지정 정보보호 인증기술연구센터 센터장
<관심분야> 암호이론, 정보이론, 정보보호