의료솔루션 사용과 관련된 효율적인 인증서 관리 시스템 설계 및 구현

(Efficient certificate management system design and implementation on the use of medical solutions)

이효승*, 오재철**

(Hyo Seung Lee, Jae Chul Oh)

요 약

현재 각 의료기관들은 의료정보화 사업을 꾸준히 진행 하고 있다. 의료기관에서 운영하고 있는 대표적인 시스템으로는 전자의무기록, 처방전달시스템 등이 있으며, 의료법에서 전자서명을 허용함으로써 의료정보를 관리함에 있어 비용절감 및 진료정보의 공유가 가능하게 되었고, 공인인증서를 활용한 의료솔루션 사용이 확산되어 가고 있다. 이러한 현실에서 인증서의 역할은 어느 무엇보다 중요하다고 할 수 있지만, 대부분 개인 인증서 관리에는 매우 적극적인 반면, 의료솔루션 등 업무와 관련된 인증서 관리에 대해서는 소홀한 것이 사실일 것이다. 업무용 인증서의 경우 업무PC에 보관하는 경우가 다수이며 이는 보안에 취약할 수밖에 없다. 이에 대한 해결책으로 인증서 서버가 존재하나 별도서버의 구축이 필요하여 중소병원의 경우 비용에 대한 부담이 적지 않을 것으로 판단된다.

본 논문에서는 추가비용을 최소화하여 별도의 인증서 서버를 구축하지 않고 인증서 파일을 BLOB로 저장하여 현재의 자원을 활용한 인증서 관리 및 보안에 효과적인 시스템을 설계 및 구현 하고자 한다.

■ 중심어 : 공인인증서 ; 전자의무기록 ; 데이터베이스 ; 인증서 서버

Abstract

Currently, different medical institutions have been carrying out the e-healthcare system project.

The system includes electronic medical record and prescription delivery system, and, the Medical Treatment law permits electronic signature for medical record management, which reduced the relevant costs and enabled sharing medical record. And medical solution using online certificates is expanding its application. In that light, the role of certificates became more important than ever. However, in contrast to active effort made to manage personal certificates, certificates related to medical solutions and other types of work are not being managed properly.

Most work-related certificates are saved in office computers, which makes them vulnerable to various security threats. Although certificate servers can be used as a solution to this problem, hospitals must build the server separately and, therefore, small and medium-size hospitals can be reluctant to bear the burden.

This study proposed a way to design and implement an effective and secure certificate management system by save the certificate file as a BLOB, using existing resources without needing to build a separate certificate server, at minimized costs.

■ keywords : Certificate ; Electronic Medical Record ; Database ; Certificate Server

Ⅰ. 서 론

IT기술의 계속되는 발전으로 의료정보분야 또한 급격한 정보화의 발전을 가져오고 있다. 이에 따라 이전까지 서류로 작성되어지던 의무기록차트가 현재 대부분의 병·의원에서 전자의무기록 시스템(Electronic Medical Record System: EMR)으로 대체되어 사용되어지고 있고, 필름을 사용하던 방사선 촬영 역시 의료 영상 저장 전송 시스템(Picture Archiving and Communication System: PACS)을 사용하고 있다. 현재 대부분의 병·의원에서 의료정보화를 꾸준히 진행 중에 있으며 더욱 확산되어가고 있는 추세이다. 이러한 기술들의 발달로 처

* 정회원, 순천대학교 컴퓨터과학과
** 정회원, 순천대학교 컴퓨터과학과

접수일자 : 2015년 11월 19일
수정일자 : 2016년 03월 14일
게재확정일 : 2016년 03월 16일
교신저자 : 오재철 e-mail : ojc@sunchon.ac.kr

방, 진료기록, 영상 등의 대부분의 데이터는 데이터베이스에 저장되어지게 되어 저장된 데이터를 효율적으로 관리하는 방안이 모색되었다. 그리하여 공인인증서를 이용한 의료 데이터 관리가 시작되었고, 공인인증서는 신원확인을 위한 수단뿐 아니라 데이터 기밀성과 무결성을 보장하는 기능이 포함되어 있고 또한 공인인증서 사용자의 행위에 대한 전자서명 기술을 사용하고 있어 부인방지 기능이 포함되어 있다[1]. 이는 공인인증서의 사용이 금융업무 뿐만 아니라 의료업무의 데이터 관리에도 정확하게 부합하는 것이라 할 수 있다.

공인인증서 사용에 관한 전자서명 법은 1999년 2월 법률 제5792호로 제정되고 그 후 계속해서 계정되어 왔으며 온라인상의 전자거래를 활성화 시키는 제도적 기반을 마련하였고, 국내 5대 공인인증기관이 지정되어 공인인증서를 발급, 관리, 운영하고 있다[2].

2013년을 기준으로 3,006만 건의 공인인증서가 발급되어 폭넓게 사용되어 지고 있고, 전자서명의 필요성이 증대되고 있다[3]. 이러한 상황임에도 불구하고 의료시스템의 사용과 관련된 공인인증서 사용자들은 인증서 저장 매체에 대한 보안 인식의 부족과, 추가적인 이동식저장장치에 대해 항상 소지하여야 한다는 불편함을 호소하고 있고, 이동식저장장치의 분실 등의 현상이 발생할 경우에도 그 피해에 대한 인식이 부족한 상태이다.

공인인증서를 활용하여야 하는 업무를 진행함에 있어 보안적인 측면과 사용자의 불편함을 해소하기 위한 방안으로 현재 인증서버라는 별도의 서버를 운영하는 방법이 있으나, 이는 별도 서버를 구축하여 운영하여야 하기 때문에 관리자로 하여금 보다 많은 관리 포인트를 요구하고 또한 중소병원 및 의원의 경우 별도의 서버구현에 대한 비용 역시 적지 않은 부담일 것이다.

본 논문에서는 기존에 사용하고 있는 데이터베이스를 활용하는 방안으로 인증서 파일을 사용자가 보유하고 있는 형태가 아닌 현재 운영 중인 DB에 BLOB 형태로 저장하여 일괄보관하는 형태로 사용자 인증 시 인증서를 해당 PC 로 다운로드하여 사용하고 해당 의료프로그램 종료 시 다운로드한 인증서를 삭제처리 하는 방식으로 공인인증서 사용자에게는 사용 시의 불편함을 최소화시키고 관리자에게는 인증서 보안등의 관리 포인트를 줄여 관리에 대한 부담을 줄이고, 최소의 비용으로 의료기관의 시스템 구현에 대한 비용적 측면을 최소화하여 삼자간의 불편함 및 부담을 줄이기 위한 방안을 모색하여 구현하고자 한다.

Ⅱ. 관련연구

1. 공인인증서

공인인증서는 전자서명의 일환으로 1998년 은행에서 먼저 시작한 공개키 기반구조(PKI)방식의 사서 인증서가 2002년 공인인증서 기반으로 전환되어 현재까지 사용되어 지고 있다[5].

가. 공인인증서의 법적근거

공인인증서라 함은 공인인증기관에서 발급한 전자서명 생성정보가 가입자에게 유일하게 속한다는 사실 등을 확인하고 이를 증명하는 전자적 정보를 말한다.(전자서명법 제 2조 7항)

문서 또는 서면에 서명, 서명날인 또는 기명날인을 요하는 경우 전자문서에 공인전자서명이 있는 때에는 이를 충족한 것으로 보고 공인전자서명이 있는 경우에는 당해 전자서명이 서명자의 서명, 서명날인 또는 기명날인이고, 당해 전자문서가 전자 서명된 후 그 내용이 변경되지 아니하였다고 추정한다.(전자서명법 제3조)

이렇게 공인인증서는 국가에서 정한 법률을 바탕으로 운영되어지며 그에 따른 책임도 물을 수 있는 것이다. 하지만 의료업에 종사하는 대부분의 사람들은 현재 의료솔루션에서 사용하고 있는 인증서가 이렇게 중요한 목적으로 사용되어 지고 있다는 사실을 인지하지 못할 정도로 정보보안의식이 결여되어 있다. 정보보안의식이란 정보보안과 관련된 사용자의 인식을 말하는 것으로 정보시스템의 사용자가 스스로 정보보안에 대한 관심이나 중요성을 인지하는 정도를 나타낸다[1].

이러한 상황에서 인증서의 보안 및 보관을 개인에게 전임하는 것은 보안 사항을 방치하는 것과 다름없는 일일 것이다.

나. 공개키 기반구조(Public Key Infrastructure)

시스템적 관점에서 공개키 기반구조는 인증기관들의 네트워크 시스템, 암호학적 키와 인증서의 배달시스템, 공개키의 인증서를 이용해 공개키를 자동으로 관리해주는 기반구조, 공개키 인증서를 발행하고 접근을 제공하는 인증서 관리 기반구조로 볼 수 있으며[7] 이는 현재 보안인증서를 발급하는 인증서 발급 시스템(CA), 보안인증서 발급신청 업무를 대행하는 인증서 등록시스템(RA), 발급된 보안인증서에 대한 유효성 검증을 수행하는 인증서 검증 시스템(OCSP), 발급된 보안인증서에 대한 효율적인 관리를 위한 인증서 관리 시스템(LDAP)으로 나눌 수 있다.

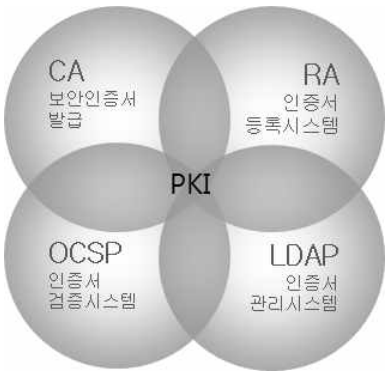

그림 1. 시스템 관점에서의 공개키 기반구조

(1) 인증기관 : CA(Certificate Authority)

사용자에게 공개 키 인증서를 발급하며 다른 인증기관에 대한 상호 인증서를 발급하는 신뢰된 개체로써 전자 서명을 생성하며 이를 인증서에 첨부하여 인증서에 대한 신뢰도를 보장해 줄 수 있다[8].

(2) 등록기관 : RA(Registration Authority)

등록기관이 공개키 기반 구조의 필수 구성요소는 아니지만 일반적으로 인증기관이 모든 인증서 소유자를 접촉할 수 없으므로 등록업무를 전담하는 등록기관을 운영한다[9].

인증서의 생성 형식으로는 X.59, WTLS, X9.68 형식이 존재하며 그중 가장 많이 사용되는 형식은 X.509형식의 인증서 이다[10].

(3) 온라인 인증서 상태 프로토콜 : OCSP(Online Certificate Status Protocol)

온라인상에서 즉시 인증서 검증이 가능하도록 하는 온라인 상태 검증 프로토콜은 IETF(Internet Engineering Task Force)가 제안하였다. 이는 중앙 집중형 서버를 이용하여 인증서의 효력정지 및 폐지 등의 상태를 실시간으로 정확히 파악할 수 있다는 특징이 있다[11].

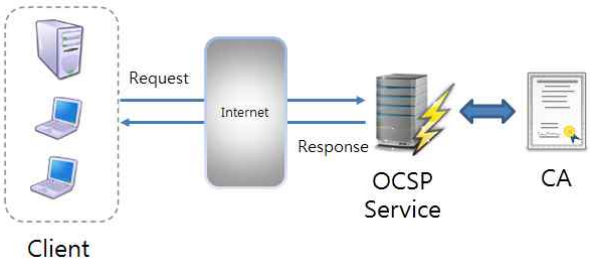

그림 2. 온라인 인증서 상태 프로토콜 서비스 구조

(4) 경량 디렉터리 액세스 프로토콜 : LDAP(Lightweight Directory Access Protocol)

LDAP는 X.500 기술의 문제를 극복하기 위해 고안된 방식으로 X.500은 디렉토리를 만들고 그에 접속하기 위한 프로토콜과 표준명세를 지원하지만 너무 크고 복잡하여 자원소모가 적고 기능이 단순화된 LDAP가 설계되었고, LDAP 디렉터리 서비스는 인트라넷 등의 네트워크 상에서 파일이나 장치 같은 자원의 위치를 찾을 수 있도록 정보를 제공하여 주고 이는 인증서 관리에도 사용되고 있다[12].

2. DBMS BLOB

BLOB란 그래픽, 음성, 텍스트, 바이너리 데이터 등을 데이터베이스에 저장하기 위한 바이트의 연속이다.

문자가 아닌 데이터 덩어리로 저장되기 때문에 내용물을 검색하거나 자료의 관리는 불가능하다.

BLOB의 종류를 구분하는 방법에는 여러 가지가 있겠으나, 저장되는 데이터의 종류에 의해서 TEXT BLOB와 BYTE BLOB로 구분이 가능하다.

TEXT BLOB의 경우 일반적인 텍스트를 저장하는 것이고 BYTE BLOB는 그래프, 목적코드, 음성, 파일등과 같은 바이너리 데이터를 저장하는 것을 의미한다[4].

본 논문에서는 DB의 BYTE BLOB 타입을 이용해 기존 사용하던 데이터베이스를 활용하여 시스템을 구성하고자 한다.

III. 인증서 관리시스템 설계

1. 인증서 관리 시스템의 구성

본 논문에서 제시할 시스템의 구성은 다음과 같다.

그림 3. 인증서 관리 시스템 구성도

기존에 사용하던 데이터베이스를 사용하여 추가적인 서버 구성이나 데이터베이스 구축이 필요 없어 비용 적인 절감 효과를 가져 올 숭 IT을 것으로 예상되며 데이터베이스를 이용하여 인증파일을 다운로드 하므로 FTP 파일전송 시스템에 비해 빠른 속도를 보장해 줄 수 있을 것으로 예상된다.

1. 인증서 관리 시스템 구조 및 운영절차

인증서 관리 시스템은 DBA와 인증서 관리자의 관리를 바탕으로 사용자가 자신의 인증서를 DB서버에 등록하고 해당 의료 시스템을 사용할 경우 자신의 인증서가 다운로드 되는 구조이다.

본 논문에서 제시할 효율적인 인증서 관리 시스템 운영의 절차는 간단한 흐름을 설명하면 다음과 같다.

① 사용자가 의료시스템에 로그인할 경우 로그인이 성공하면 하면 인증서 가져오기 옵션이 선택되었는지 확인한다.

② 해당 옵션이 선택되어 있는 경우 로그인한 사용자의 ID를 바탕으로 데이터베이스에 해당 인증서가 존재하는지 확인한다.

③ 해당 사용자의 ID가 인증서 관리 테이블에 존재할 경우

BLOB의 데이터를 파일로 변환하여 컴퓨터의 인증서 저장 위치에 저장한다.

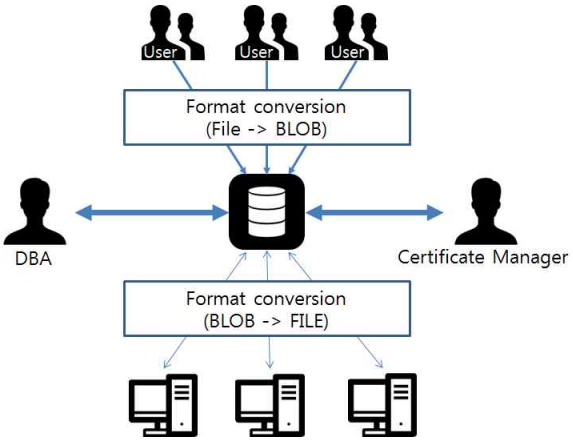

그림 4. 인증서 관리 시스템 구조

④ 저장된 인증서를 바탕으로 인증서 로그인을 하고 의료시스템 업무를 수행한다.

⑤ 해당 업무를 종료하고 프로그램을 종료 시킬 경우 기존에 다운받았던 인증서를 자동 삭제 시킨다.

이러한 방식으로 해당 인증서는 업무를 사용할 때 다운받아 사용하고, 사용이 끝나고 업무를 종료할 경우 해당인증서를 삭제하여 인증서 사용에 있어서 인증서 보관에 대한 부담을 줄일 수 있는 장점이 있고, 그로인하여 물리적인 보안 측면에서도 효율적일 것으로 예상된다.

2. 인증서 관리 시스템 테이블 설계

인증서 관리를 위해 의료기관에서 사용 중인 데이터베이스를 활용하여 최소한의 비용으로 효율적인 인증서관리 시스템을 만들고자 하는 것이 본 논문의 취지이고, 관리자로 하여금 최소한의 관리 포인트로 운영할 수 있게 하여야 하기 때문에 Master Table, Sub Table 이 두 개의 테이블을 이용하여 인증서 관리 시스템을 설계 하고자 한다.

표 1. Master Table 설계

Column_name	Data_type	Comments
NANM	VARCHAR2(50)	명칭
FOLD	VARCHAR2(200)	폴더명
STDT	VARCHAR2(8)	시작일
ETDT	VARCHAR2(8)	종료일
ISDT	VARCHAR2(14)	접속일시
LEVL	VARCHAR2(50)	법인/개인 구분
ORGA	VARCHAR2(20)	발급처
SRNO	VARCHAR2(30)	일련번호
USFG	VARCHAR2(2)	사용유무
SANO	VARCHAR2(20)	사용자코드

Master Table은 인증서의 Title를 기록하여 사용자별 인증서를 구분할 수 있는 테이블로 Master Table의 각 칼럼은 다음과 같다.

① NANM : 해당 인증서의 사용처나 용도를 기입

② FOLD : 인증서의 주체 필드로 이후 인증서를 해당 컴퓨터에 저장 할 경우 디렉터리 명으로 활용

③ STDT : 인증서의 유효 시작일로 해당 인증서의 실제 사용가능한 시작일자

④ ETDT : 인증서의 유효 종료일로 해당 인증서의 종료일자

⑤ ISDT : 인증서를 이용하여 마지막으로 접속을 시도일자

⑥ LEVL : 인증서의 구분으로 개인, 법인(범용)을 구분

⑦ ORGA : 해당 인증서의 발급기관

⑧ SRNO : 해당 인증서의 일련번호를 기입

⑨ USFG : 사용자 퇴사 등의 사유로 인하여 관리자가 강제로 해당 인증서의 사용과 종료를 구분

⑩ SANO : 인증서의 실제 소유자로 로그인시 해당 컬럼의 내용을 비교하여 다운 받을 인증서를 선택

표 2. Sub Table 설계

Column_name	Data_type	Comments
SRNO	VARCHAR2(20)	일련번호
FINO	NUMBER	파일번호
FINM	VARCHAR2(200)	파일이름
FIDT	VARCHAR2(8)	파일생성일
FISZ	NUMBER	파일사이즈
FIPS	VARCHAR2(200)	파일경로
FIBL	BLOB	실제파일

Sub Table에는 인증서의 Title를 기록하고 각 칼럼은 다음과 같다.

① SRNO : Master Table의 SRNO 칼럼과 비교하여 사용하기위한 킷값으로 인증서 파일의 구분

② FINO: 인증서 각각 파일들의 파일번호를 지정

③ FINM: 각 파일의 이름을 기입한다. 이후 컴퓨터에 저장할 때 해당 파일 이름으로 파일을 생성

④ FIDT: 파일의 생성일을 기입

⑤ FISZ: 파일의 사이즈를 기입

⑥ FIPS: 파일경로를 지정한다. 단, 컴퓨터에 저장할 시 OS의 버전에 따라 다르게 설정되어야 할 경우가 존재하므로 상대경로에서의 현재경로 “.” 으로 통일

⑦ FIBL: 실제 파일을 BLOB 형식으로 변환하여 저장하기 위한 컬럼

표1, 표2의 테이블 설계를 바탕으로 기준 자료를 마스터 테이블에 저장하고 실제 각각의 파일들은 서브테이블에 저장하

여 운영하는 방식을 이용하고자 한다.

IV. 인증서 관리시스템 구현

본 논문은 일반상황에서는 인증서를 보관하고 있지 않은 상태에서 의료솔루션에 로그인 할 때 필요한 인증서을 DB에서 다운받아 사용하고 이후 프로그램을 종료할 경우 다운받았던 인증서를 선택 삭제하는 방식의 인증서 관리 시스템을 구현하고자 하고 있다.

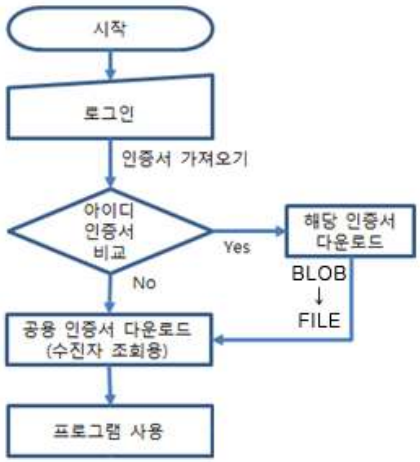

그림 5. 프로그램 사용 인증서 다운로드 알고리즘

인증서 관리 시스템을 구현하기 위해 Master Table과 Sub Table에 테스트를 위한 인증서를 저장하기 위하여 적용한 데이터는 표 3과 표 4와 같다.

표 3. Master Table Sample Data

Column_name	Data
NANM	수진자조회
FOLD	cn=테스트인증서1234567890ABCDEFGHIJ,ou=건강보험,ou=MOHW RA센터,ou=등록기관,ou=licensedCA,o=KICA,c=KR
STDT	20150717
ETDT	20160801
ISDT	20151116240143
LEVL	범용(법인)
ORGA	KICA
SRNO	031ff7be
USFG	Y
SANO	ALL

표 4 Sub Table Sample Data

Column_name	Data
SRNO	031ff7be
FINO	1,2,3,4,5
FINM	CaPubs kmCert.der kmPri.key signCert.der signPri.key
FIDT	20150717160432
FISZ	2444
FIPS	.
FIBL	BLOB NOT RETRIEVED

표3, 표4와 같이 인증서의 정보는 Master Table에 모두 저장하고 Sub Table에는 인증서를 이루는 각각의 CaPubs, kmCert.der, kmPri.key, signCert.der, signPri.key 파일을 BLOB형식으로 저장한다.

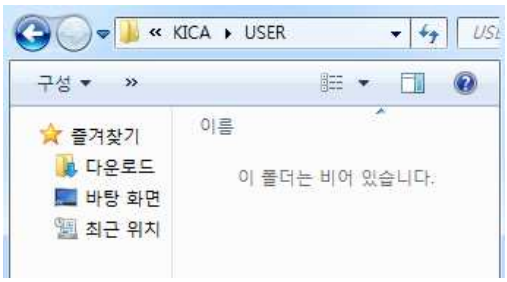

그림 6. 로그인 전 인증서 위치 화면

그림6 에서와 같이 로그인 전의 인증서 위치에는 인증서가 존재하지 않는 상태임을 확인 할 수 있다.

그림 7. 의료솔루션 통합로그인 화면

그림7 에서와 같이 로그인 시 인증서를 가져오기 위해서 “인증서가져오기” 체크박스를 선택한 상태에서 로그인을 하면 사용자를 비교하여 인증서의 유무를 판단하고 그에 따른 이벤트 및 함수를 호출하여 해당 인증서를 다운받게 된다.

BLOB를 파일로 변환하기 이전에 프로그램 종료 시 해당 인증서에 대한 정보를 보관하기 위한 소스는 다음과 같다.

```
String ls_srno, ls_orga, ls_fold, ls_sano
Long ll_dw1cnt, ll_dw2cnt
Integer i, j, li_cnt
dw_abcerm_dn.accepttext( )
li_cnt = 0
ll_dw1cnt = dw_abcerm_dn.rowcount( )
for i = 1 to ll_dw1cnt
    ls_srno = dw_abcerm_dn.GetItemString(i,'srno')
    ls_orga = dw_abcerm_dn.GetItemString(i,'orga')
    ls_fold = dw_abcerm_dn.GetItemString(i,'fold')
    ls_sano = dw_abcerm_dn.GetItemString(i,'sano')
    if ls_sano <> 'ALL' then
        li_cnt = li_cnt + 1
        gs_certlist[li_cnt] = "C:\Program Files\NPKI\"+ls_orga+"\USER\"+ls_fold
    end if
    dw_2.event ue_retrieve_info(ls_srno)
    dw_2.accepttext( )
    ll_dw2cnt = dw_2.rowcount( )
        hpb_1.maxposition = ll_dw2cnt
    for j = 1 to ll_dw2cnt
        of_filedownloadblob(j, true, ls_fold, ls_orga)
        hpb_1.position = j
    next
next
```

위의 소스에서 gs_certlist 배열에 다운로드한 인증서의 정보를 보관하고 프로그램 종료 시 해당 인증서를 삭제한다.

실제 BLOB를 파일로 변환하여 해당 디렉터리로 다운로드하는 함수의 내용은 다음과 같다.

```
STRING ls_path, ls_target, ls_sdt
string ls_data, ls_tmp, ls_file, ls_Write
long lrow, lrowcnt
uLong lul_sLen, lul_tLen
blob lb_data
Boolean lb_Copy = false
datetime ldt_tdt
String ls_code
long ll_FILENUMB
n_filesys ln_fs
nvo_blobtostring ln_bs
if Not DirectoryExists("C:\Program Files\NPKI\") then
    CreateDirectory("C:\Program Files\NPKI\")
end if
if Not DirectoryExists("C:\Program Files\NPKI\"+as_orga+'\') then
    CreateDirectory("C:\Program Files\NPKI\"+as_orga+'\')
end if
if Not DirectoryExists("C:\Program Files\NPKI\"+as_orga+"\USER\") then
    CreateDirectory("C:\Program Files\NPKI\"+as_orga+"\USER\")
end if
ls_target = as_orga+'\USER\'+as_fold
    ls_path = "C:\Program Files\NPKI\" + ls_target + '\'
if Not DirectoryExists(ls_path) then
    CreateDirectory( ls_path )
end if
ls_file = dw_2.GetItemString(row, "finm")
ls_data = ''
ls_Write = ls_path + ls_file
if ab_Check then
    ls_sdt = dw_2.GetItemString(row, "fidt")
    ln_fs.of_GetFilewriteDate(ls_Write, ldt_tdt)
    if ls_sdt > String(ldt_tdt,'YYYYMMDDHHMMSS') then lb_Copy = true
    lul_sLen = dw_2.GetItemNumber(row, "fisz")
    lul_tLen = FileLength( ls_Write )
    if lul_sLen > lul_tLen then lb_Copy = true
end if
if lb_Copy then
    ls_code = dw_2.GetItemString(row, "srno")
    ll_FILENUMB = dw_2.GetItemNumber(row, "fino")

    of_helpdisp( ls_file+' Load 중... ',0)
    SELECTBLOB fibl
    INTO :lb_data
    FROM abcerh
    WHERE srno = :ls_code
```

```
        AND         fino = :ll_FILENUMB
        USING       SQLCA;
        yield()
        if Len(lb_data) > 0 then
            of_helpdisp( ls_file+' 파일 Writing ... ', 0)
            ln_fs.of_FileWrite( ls_Write, lb_data )
        end if
    end if
    return 0
```

위의 소스에서 확인할 수 있듯이 인증서의 경로를 생성해주고 그 후 BLOB형식의 데이터를 파일로 변환하여 해당 경로에 저장하는 형식으로 구현되어 있다.

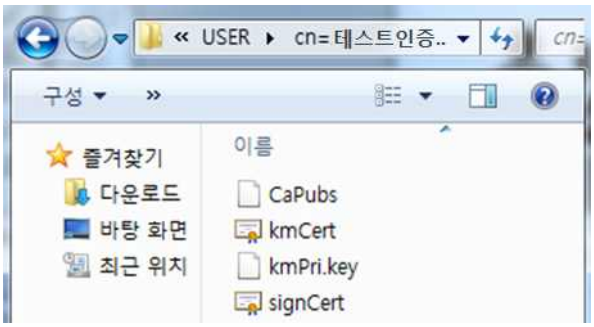

그림 8. 통합로그인 후 인증서 위치 화면

로그인 후 인증서의 경로를 확인해 보면 그림8 에서와 같이 파일들이 저장되어 있는 것을 확인할 수 있다.

마지막으로 프로그램 종료 시 다운로드 받았던 인증서를 삭제하기 위한 알고리즘과 함수의 내용은 다음과 같다.

해당 인증서 파일을 삭제하기 위해 로그인 시 다운로드 하였던 인증서 Master 값을 프로그램 상에 저장한 후 프로그램 종료 시 해당 사용자의 인증서를 확인하여 삭제한다.

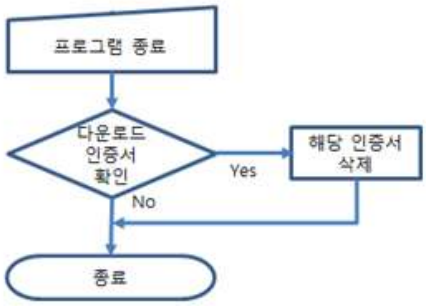

그림 9. 사용 인증서 삭제 알고리즘

```
nvo_filesys in_fs
String ls_file, ls_path, ls_name[]
Integer li_del, i, j, icnt
boolean lb_subdir[]

for i = 1 to upperbound(gs_certlist)
    ls_path = gs_certlist[i]
    icnt = in_fs.of_getfiles(ls_path, ls_name[],lb_subdir[])

        for j = 1 to icnt
            ls_file = ls_path +'\'+ ls_name[j]
            FileDelete(ls_file)
        next
    li_del = RemoveDirectory (ls_path)
    yield()
next
return li_del
```

이렇게 하여 기존의 데이터베이스에 테이블 두 개만을 추가 생성하여 효율적인 인증서 관리 시스템을 구현하였다.

표 5. 기존 인증서 관리 방안과의 비교

구분	일반관리	인증서버	제안시스템
관리대상자	개별관리	인증담당자	인증담당자
저장장치	업무용PC 이동디스크	별도인증 서버	기사용중인 DB서버
저장형태	FILE	FILE	BLOB
사용자편의	저	고	고
시스템비용	저	고	저

III. 결 론

현재 사용되어지고 있는 인증서버의 경우 서버구축 및 그에 따른 비용 부담이 적지 않고 파일형식의 인증서 저장 방식을 적용하고 있기 때문에 인증 서버에 원하지 않는 사용자가 접근 할 경우 보안상의 문제가 발생할 수도 있을 것이다.

이를 방지하기 위해 기존에 의료기관에서 사용 중인 데이터베이스를 이용하여 비용을 최소화 하고 파일을 DB화 하여 BLOB 칼럼에 저장함으로써 인증서에 대한 보안을 강화하고, 또한 이동식저장장치를 가지고 다니는 불편함을 해소할 수 있으며, 컴퓨터 하드디스크에 인증서를 저장하여 발생할 수 있는 보안 사고를 미연에 방지할 수 있을 것으로 생각된다.

단, 모든 의료 프로그램에 적용하기 위해서는 각 의료프로그램과 연동작업이 필요할 것으로 판단되며 이러한 사항만 충족된다고 하였을 경우 최소비용으로 사용자 및 관리자의 업무 효율성 등에 최대효과를 가져올 수 있을 것이라 생각된다.

앞으로 본 논문에서 구현한 효율적인 인증서 관리 시스템을 실제 병·의원에 적용시켜 보고 그에 따른 문제점 및 발전방향에 대하여 계속된 연구를 진행하고자 한다.

References

[1] 김세윤, 김태성, “공인인증서의 신뢰 및 지속사용 의도에 관한 연구”, *한국경영정보학회 학술대회논문집*, Vol.2014 No.1 pp.1101-1105

[2] 박영진, 김선종, 이동훈, “인증서와 개인키 유출 방지를 위한 보안키 저장소 Secure Key Store”, *정보보호학회논문지*, 제24권 제1호, pp.31-40, 2014

[3] 조상래, 조영섭, 김수형, 노종혁, 최대선, 진승헌, “터치사인 인증서 관리 시스템 구현”, *대한전자공학회 하계종합학술대회 논문집*, Vol.2014 No.6 pp.598-601

[4] 최기호, 강현철, “멀티미디어 데이터 관리를 위한 다양한 BLOB 타입의 분류 체계”, *한국정보과학회 1993년도 봄 학술발표논문집*, 제20권 제1호, 1993. pp.73-76

[5] 김종열, “공인인증서와 바이오 정보를 이용한 강화된 인증 기법”, *숭실대학교 석사학위 논문*, 2012. 2

[6] 홍시환, “공인인증서 기반의 전자지불시스템 구축”, *동아대학교 석사학위 논문*, 2000. 2

[7] 이용규, 남승필, 박용우, “공개키 기반구조의 개념 정립과 추진체계”, *국가정책연구(Public policy review)*, Vol.12 No.- pp.71-88

[8] 홍성옥, “분산 OCSP에서 인증서 상태 검증을 위한 효율적인 기법”, *동국대학교 대학원 박사학위 논문*, 2004. 8

[9] 김성덕, “공개키 인증서의 효율적인 발급 및 이용 방법에 관한 연구”, *성균관대학교 박사학위 논문*, 2006. 8

[10] C. Adams, S. Farrell. RFC 2510 "Internet X.509 Public Key Infrastructure Certificate Management Protocols", March 1999

[11] 최승권, 장윤식, 지홍일, 신승수, 조용환, “응답시간 단축을 위한 분산 OCSP 인증서 검증 모델”, 한국통신학회논문지, Vol.30 No.4A, pp.304-31, 2005

[12] 문남두, 안건태, 김진홍, 이명준, “그룹통신을 이용한 견고한 LDAP 서버의 설계 및 구현”, *한국정보과학회 학술발표논문집*, Vol.28 No.1A, pp.430-432

저 자 소 개

이효승(정회원)

2005년 동국대학교 정보통신공학과 학사 졸업.
2008년 순천대학교 정보통신공학과 석사 졸업.
2014년~ 현재 순천대학교 컴퓨터과학과 박사 과정
2013~ 현재 청암대학교 컴퓨터정보과 겸임교수
<주관심분야 : 의료정보시스템, u-헬스케어 시스템>

오재철(정회원)

1978년 전북대학교 전기공학과 (공학사)
1982년 전북대학교 컴퓨터공학과 (공학석사)
1998년 전북대학교 컴퓨터공학과 (공학박사)
1984년~1986년 기전대학교 전자계산학과 전임강사
1986년~현재 순천대학교 컴퓨터공학과 교수
<주관심분야 : 임베디드시스템, USN, 네트워크 설계 및 분석>