한국산학기술학회논문지
Vol. 9, No. 2, pp. 370-378, 2008

공인전자문서보관소를 위한 인증서 기반의 안전한 전자문서 전송시스템 설계 및 구현

김대중[1*], 김정재[1], 이승민[2], 전문석[3]

Design and Implementation of a Secure E-Document Transmission System based Certificate for CEDA (Certified E-Document Authority)

Kim Dae Jung[1*], Kim Jung Jae[1], Lee Seung Min[2] and Jun Moon Seog[3]

요 약 공인전자문서보관소는 전자문서 보관의 법적 효력을 부여해 문서를 안전하게 보관하고, 전자문서의 내용과 송수신 여부 등을 증명해 주는 신뢰할 수 있는 제3의 기관을 말한다. 본 논문에서는 공인전자문서보관소 구축 핵심 기능인 송수신 보안 기능에 초점을 맞추고 있으며, 공인전자문서보관소 업무 서버와 이용자 간에 각 시스템간의 데이터 송수신시 안전하게 메시지를 보낼 수 있도록 공개키를 이용한 암호화 시스템을 제안 및 구현하였다. ITU-T의 X.509 기본 원칙을 준수하며, 대량의 데이터를 속도 향상을 위해 대칭키 암호화 알고리즘을 사용하였으며, 여기에 사용된 키만을 개인키로 암호화 시켰으며, 데이터의 무결성을 위해 메시지를 전자서명을 통해 수정하는 것을 방지하였다. 또한 시스템간의 인증을 하기 위해 인증서를 통해 서로 인증할 수 있도록 하였다.

Abstract The CEDA(Certified E-Document Authority) is a reliable third party that deposit electronic document having legal effects securely, and verify contents of document or transmission. This paper focuses on a function of secure transmission among several important functions, and implements public key encryption system for secure transmission when server and user communicate for image transmission. This paper follows a standard fundamental rule of X.509 in ITU-T, and it uses symmetric encryption algorithm to raise speed of a large data operation. A key of symmetric encryption algorithm is encrypted by private key in public key system, it protects to be modified using digital signature for data integrity. Also it uses certificates for mutual authentication.

Key Words : Information, 네트워크, Model, Structure, 컴퓨터통신, Mutual authentication, SOAP

1. 서론

우리나라 기업 · 금융기관 등은 각종 문서 또는 서류의 유통 · 보관에 연간 1조 원 이상을 소요하고 있는 것으로 추정되며, 검색 · 참조 등 보관문서의 활용도 어려움이 있는 것이 현실이다. 이러한 종이문서를 전자문서로 대체하면 종이문서 보관에 필요한 문서 창고를 점진적으로 감축할 수 있게 됨은 물론 검색 · 활용이 온라인상에서 가능하게 되어 시간과 비용을 획기적으로 절약할 수 있다[1],[2]. 2005년 3월에 전자거래기본법이 개정된 이래, 2007년 8월 「공전소 이용자시스템 기술규격 V1.1」 제정까지 전자문서를 위한 법률적 기반을 제공하여 전자문서 이용 확산을 통한 비용절감과 업무를 효율화 했다. 본 연구는 공인전자문서보관소를 구축하는데 있어, 핵심 서비스중의 하나인 공인전자문서보관소와 이용자간 전자문서의 안전한 송수신 설계 및 구현에 있으며, 한국전자거래진흥원의 「이용자시스템과 공인전자문서 보관소 간 연계인터페이스 기술 규격 V1.10」 중 송수신 인터페이스 규약에 부합하도록 노력하였다. 본 논문의 구성은 다음과 같다. 2장은 관련연구로서 공인전자문서보관소에 대한 전반적인 개념, 암호화 방법 및 해외 전자문서보관소 관

[1]숭실대학교 컴퓨터학과(박사과정)
[2](주)리테일테크 기술연구소 수석연구원
[3]숭실대학교 컴퓨터학과(부교수)
*교신저자: 김대중(djkim@fsb.or.kr)

련 비즈니스 모델을 분석하고, 3장에서는 공인전자문서 보관소와 이용자 시스템간의 안전한 연계서비스 구현방안 및 제안하는 시스템에 대한 암호화 과정, 4장에서는 구현 방법, 5장에서는 실험평가, 6장에서는 결론 및 향후 연구 방향을 제시한다.

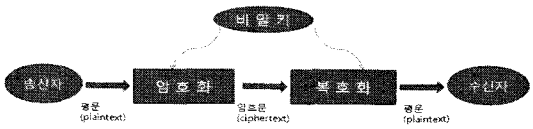
2. 관련 연구

2.1. 암호시스템

암호시스템은 키 관리 형태에 따라 비밀키 암호 시스템과 공개키 암호 시스템으로 분류 될 수 있다.

2.1.1 비밀키 암호 시스템(Secret-Key Cryptosystem)

비밀키 암호 시스템은 암호화와 복호화에 같은키를 사용함으로서 대칭키 암호시스템(Symm etric-Key Cryptosystem)이라고도 불린다. 송신자는 전송하고자 하는 평문(Plaintext)을 키와 암호화 알고리즘을 통해 암호문(Cypertext)으로 변환하여 수신자에게 전송하고, 수신자는 같은 키를 사용하여 원래의 평문을 생성한다. [그림 1]은 비밀 키 암호 시스템의 수행 과정을 보인다. 비밀키 암호 시스템은 암호화 및 복호화 속도가 빠른 장점이 있는데 반면에 키 관리가 어렵다는 단점을 가진다. 이러한 문제로 인해 대칭키 암호시스템은 주로 데이터의 기밀성을 보장하기 위한 일반적인 암호화에 사용된다[3].

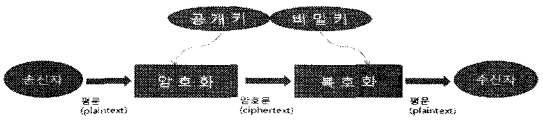


[그림 1] 비밀키 암호 시스템

비밀키 암호시스템은 SSL (Secure Soket Layer)[4]이나 IPSec (IP Security Protocol)[5]등 보안 프로토콜 등에서 중요한 역할을 하며 대표적인 비밀키 암호 알고리즘으로는 DES(Data Encryption Standard)[6], IDEA (international Data Encryption Algorithm), RC2 & RC4, AES(Advanced Encryption Standard) 등이 있다.

2.1.2 공개키 암호 시스템(Public-Key Cryptosystem)

공개 키 암호 시스템에서는 하나의 키는 모든 사용자들에게 공개하며, 다른 하나는 당사자 자신이 비밀로 보유한다. 이 때 공개하는 키를 공개키라고 하며 비밀로 보유하는 키를 비밀키라고 한다.



[그림 2] 공개키 암호 시스템

[그림 2]는 공개키 암호 시스템의 수행 과정을 보인다.

공개키 암호 시스템은 공개키를 디렉토리를 통해 알려주기 때문에 비밀 키 암호 시스템에서 요구되는 안전한 키 분배가 필요 없다는 장점을 가지는 반면, 비밀 키 암호 시스템에 비해 키의 길이가 길고 알고리즘 수행 속도가 매우 느리기 때문에 긴 평문을 암호화 하는 데에는 부적절하다는 단점을 가진다. 따라서 디지털 서명은 공개키 암호 시스템을, 평문의 암호화는 비밀 키 암호시스템을 사용하는 복합 암호 시스템이 널리 사용되고 있다. 대표적인 공개 키 암호화 알고리즘으로는 RSA(Rivest-Shamir-Adleman)방식[7], Rabin 방식, ElGamel 암호 방식[8], Diffie-Hellman 방식, DSA(Digital Signature Algorithm), ECC(Elliptic Curve Cryptosystem) 등이 있다.

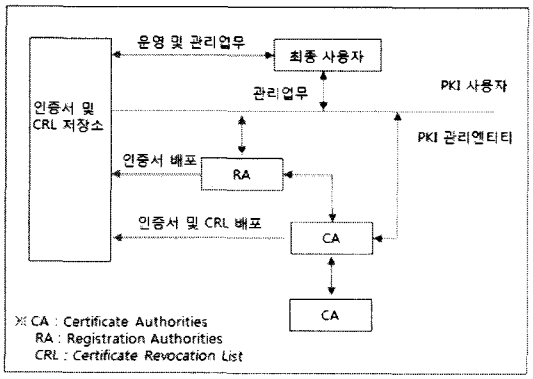
2.1.3 디지털 서명(Digital Signature)

디지털 서명은 사용자 인증, 메시지 인증, 부인 방지 서비스의 구현을 위한 대표적인 기술이다. 디지털 서명 방식은 크게 2가지 방식으로 분류 될 수 있다. 하나는 비밀 키로 암호화해서 공개키로 복호화하면 본래의 메시지가 복호화되는 메시지 복원형 디지털 방식이고, 다른 하나는 해쉬 함수를 이용해서 그 결과 값에 디지털 서명을 생성하고 검증 시 서명을 복호화해서 해쉬 값과 비교하는 부가형 디지털 서명 방식이다. 디지털 서명의 종류로는 1991년 미국에서 개발한 표준 디지털 서명인 DSS (Digital Signature Standard)를 비롯하여, 독일의 Schorr 방식, NyBerg-Rueppel 방식 등이 있고, 공개 키 암호 시스템을 이용한 Rabin, RSA, Elgamel 서명 방식, ID를 이용한 Flat-Shamir 방식과 Ohta 방식, Knapsack 문제를 이용한 Merkle-Hellman 방식등의 다영한 디지털 서명 방식이 사용된다[9].

2.2 PKI 암호화 시스템

PKI는 공개키 암호 시스템과 공개키에 대한 인증서를 기반으로 보안기능을 제공하는 정보보호 기반구조이다. 인증서비스의 기반기술로는 현재 사실상의 표준으로 받아들여지고 있는 ITU-T의 X.509가 있다. X.509를 이용한

기본적인 시스템 구성은 [그림 3]과 같다.



[그림 3] X.509 시스템 구성

공개키 기반 구조는 개방형 네트워크에서 안전한 서비스가 이루어질 수 있도록 통신 정보의 비밀성, 인증, 무결성, 부인방지 등의 기본적인 보안 서비스를 가장 효과적으로 제공하는 기반 구조이다. 공개키 암호 기술의 문제점은 공개키의 가용성을 훼손하는 경우에 발생한다. 공개키의 가용성이란 어느 누구든지 다른 사용자의 공개키가 필요한 경우에 이를 사용할 수 있는 서비스이다.

공개키가 위·변조되지 않았음을 보장하는 문제 즉, 공개키의 무결성을 보장하기 위해 공개키 대신 공개키와 그 공개키의 소유자를 연결하여 주는 인증서(Certificate)를 공개하고, 인증서는 신뢰할 수 있는 제3자인 인증기관이 자신의 개인키로 서명하여 공개키를 인증하는 시스템을 PKI 시스템이라 한다.

PKI 시스템은 공개키 암호기술이 안전하게 적용될 수 있는 기반구조로써 공개키와 그 소유자를 연결해 주는 인증서, 키와 인증서를 안전하게 관리해주는 서비스, 그리고 인증서의 유효성 여부를 확인할 수 있는 구조라고 정의한다[10].

2.3 공인전자문서보관소

2.3.1 추진 배경 및 역할

전자거래의 확산으로 전자문서 이용이 증가하고 있지만 종이문서 활용 또한 병행되고 있다. 전자거래기본법은 전자문서에 종이문서와 동일한 효력을 부여하고 있으나 전자문서는 위·변조 여부를 확인하기 어려워 전자문서 활성화에 지연이 되었으며, 이로 인해 기업프로세스의 e-비즈니스화 지연 및 종이문서 생산·유통·보관에 많은 비용이 발생되고 있다. 2005년 3월 전자거래기본법 개정으로 신뢰할 수 있는 제3의 기관으로 이용자로부터 전자문서를 받아 진본성을 유지하고 보관하며, 이용자 요청 시 불변경 등 법적 효력이 적용된 전자문서와 증명서를 발급해주는 공인전자문서보관소 구축의 발판이 마련되었다.

2.3.2 주요 서비스

공인전자문서보관소는 크게 보존, 송수신, 증명서비스를 제공한다[11]. 보존서비스는 문서가 생성되어 폐기될 때까지의 안전한 보존을 목적으로 하여 문서작성, 문서변환, 문서관리, 문서검색, 문서열람 그리고 백업과 복구 등의 기능을 제공한다. 송수신 서비스는 보존된 문서를 유통시키거나 유통만을 위해 제출된 문서에 대해 지정된 수신자에게 문서를 배달하는 서비스로서 문서의 배달, 송수신 확인, 암호화 연동 등의 기능을 제공한다. 증명서비스는 문서의 등록증명, 배달증명, 내용증명, 원본성 증명 그리고 증명서발급 등의 기능으로 구성된다. 이러한 세 가지 기본서비스에 더하여 기존 종이문서의 전자화를 위한 스캔 서비스와 사용자가 보관소를 이용할 수 있도록 하는 웹 인터페이스도 제공한다. 또한 공인인증기관과 TSA(Time Stamp Authority)와 연계되어 사용자 인증과 증명서비스를 위해 시점 정보 서비스도 할 수 있다.

2.3.3 해외 전자문서보관소 관련 비즈니스 모델 분석

현재 국내에서 추진 중인 전자문서보관소 비즈니스 모델인 보관서비스, 송수신 서비스, 증명서비스와 같은 기본 서비스와 연관된 비즈니스 모델에 대해 설명한다[12].

① 보관서비스

가. Iron Mountain(미국)의 디지털 보존은 전자 문서에 대하여 안전하고 합법적이며 저비용으로, 장기간 보존에 적합하게 설계된 혁신적인 저장 솔루션을 제공하며, 현재 종이문서와 전자문서에 동일한 서비스를 제공하며 스케닝 및 디지타이징 등에 적용 가능하도록 설계되어져 있다.

나. RICOH Trusty Cabinet(일본)의 통합 시스템 모델은 사용자의 전자문서를 업무 프로세스 시스템을 통해 TrustyCabinet에 보관하는 서비스를 제공하고 있다.

다. 영국정부기록보존소는 국가 보관소로서, 전자기록을 영구 기록으로 보관소에 저장(archiving)하며 기업이나 정부를 대상으로 서비스를 제공하고 있다.

라. OAIS(Open Archival Information System) 참조모델은 ISO의 요청으로 국제협의체인 CCSDS에서 개발하였으며, ISO의 표준 모델로 채택되었다.

② 송수신 서비스

일본의 Construction-ec.com사는 기업간 전자계약서 등의 교환보관 등을 실시하기 위해 전자서명, 장기원본 보관, 원본성 증명 등의 서비스를 제공한다.

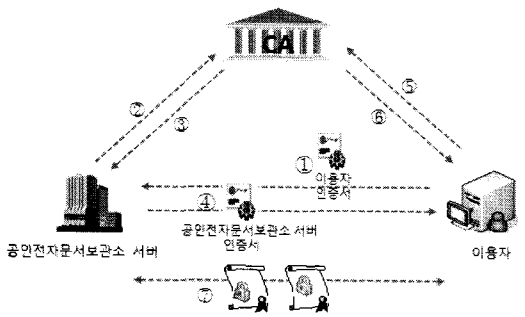
③ 증명서비스

미국의 Surety는 해쉬함수와 타임스템프의 기능을 이용하여 특정일시에 공증한 디지털 정보의 변경 및 누설의 유무를 확인하는 서비스를 제공하며, NTT DATA(일본)사의 SecureSeal은 미국의 Surety의 기술을 활용하여 전자문서의 원본성을 제3자가 기술적으로 증명하는 서비스를 제공한다.

3. 제안 시스템

3.1 제안 시스템의 구조

다음 [그림 4]는 제안하는 시스템 구조이며, 2007년 8월 「공전소 이용자시스템 기술규격 V1.1」에 제시된 연계인터페이스 기술규격을 준수한다.

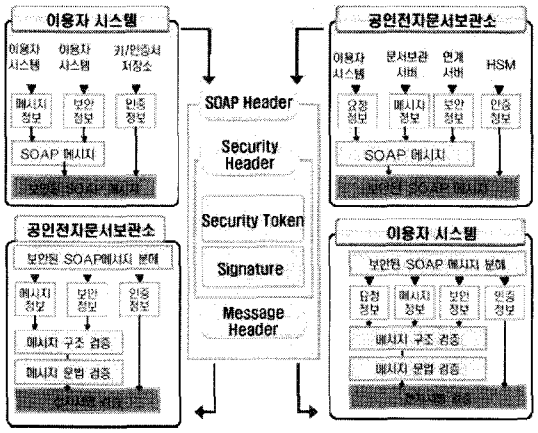


[그림 4] 제안하는 시스템의 구조

① 이용자 인증서를 공인전자문서보관소 서버로 전송한다.
② 이용자 인증서 상태 검증요청을 한다.
③ 이용자 인증서 상태를 검증한다.
④ 공인전자문서보관소 서버 인증서를 이용자에게 전송한다.
⑤ 공인전자문서보관소 서버 인증서 상태를 검증 요청한다.
⑥ 공인전자문서보관소 서버 인증서 상태를 검증한다.
⑦ 인증완료시 전달 DATA는 대칭키와 공개키를 이용하여 암호화한다.

3.2 연계 인터페이스 구현 방안

데이터 전송은 기본적으로 SOAP(Simple Object Access Protocol)을 통해 전송이 된다. 따라 본 메시지 역시 모두 SOAP을 통해 전송이 되며[13], 세부적인 연계 모듈의 구조는 아래 [그림 5]와 같다.



[그림 5] 연계 파트너 모듈의 구조

① 정보 패키지 준수

웹서비스로 송수신하는 모든 문서 정보는 공인전자문서보관소 기술규격 '보관문서 속성 정보 관리를 위한 정보패키지 표준화' 에 따라 생성한다.

② 송수신 문서 보안

WS-Security V1.0 규약 기반의 보안을 적용하며[14], 메시지 무결성, 송신자 인증, 송수신 부인방지를 위한 전자서명을 지원하고, 메시지 기밀성을 위한 암호화 또한 지원한다.

③ 기본 제공 인터페이스

문서관리를 위한 기본 기능으로 등록, 검색, 발급, 문서 3자 발급, 증명서를 이용한 발급, 이관, 폐기, 보관 연장 등이 있으며, 증명서 관련 기능으로는 발급, 갱신 발급, 검증, 다운로드 기능을 가진다.

④ 신뢰 메세징 처리

전송 오류 시 자동 재 전송 기능과 중복 메시지 자동 인지 기능이 있다.

⑤ 성능

대용량 메시지 처리 기능 및 멀티쓰레드 처리 등과 같은 성능향상을 위한 기능을 가지고 있다[15].

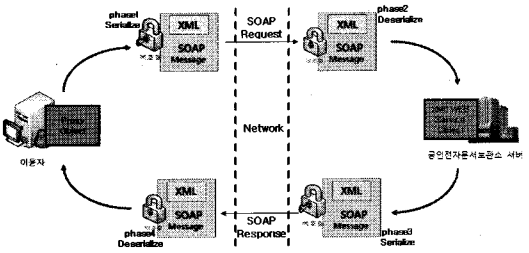
3.3 메시지 암호화 과정

공인전자문서보관소와 이용자 간에는 인터넷 구간을 통하여 자료가 오가므로 서로 주고받는 데이터는 신용할 수 없는 데이터이기 때문에 시스템간의 인증과정과 중간의 공격자에 의해 전송한 메시지의 무결성을 검증해야 한다. 본 논문에서 제안하는 암호화 메시지를 통해 시스템간의 인증 및 무결성, 데이터 암호를 모두 할 수 있는 방법을 제시하며, 다음 [그림 6]과 같다.

공인전자문서보관소 업무처리 서버를 "Server"라 칭하고, 이용자 PC를 "Client"라 칭하면 Client에서 Server에게 메시지를 전송하기 위한 각 단계별 프로세스는 다음과 같다.

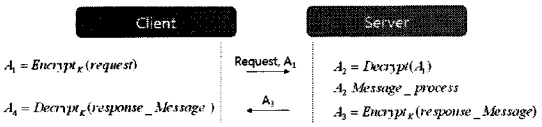
① Server는 랜덤 넘버 R_1을 생성한다.
② R_1은 Server의 비밀키로 전자서명된 후, Client의 공개키를 이용하여 암호화(T_1)한다.
③ T_1의 값을 SOAP 메시지로 Client에게 전송한다.
④ Client는 전송된 T_1을 복호화 하여 서명된 R_1을 얻고, 이 데이터가 올바른 데이터 인지 검증한다.
⑤ Client는 새로운 랜덤 넘버 R2를 생성 후, R'_1와 R_2를 연접(concatenation)연산한 데이터를 Client의 비밀키로 전자서명후 Server 공개키로 암호화(T_4)한다.
⑥ T_4의 값을 SOAP 메시지로 Server에게 전송한다.
⑦ Server는 전송된 T_4를 복호화 하여 서명된 T_4를 얻고, 전자서명을 검증하여 $R'_1 \| R_2$ 를 도출한다.
⑧ $R'_1 \| R_2$ 중에 R'_1 을 따로 분리하여 Server가 보유한 R_1와 비교 검증한다.
⑨ 검증 후 R'_1 와 분리된 나머지 부분인 R'_2를 세션키 $K(R_1+R_2)$로 암호화(T_7)한다.
⑩ T_7의 값을 SOAP 메시지로 Client에게 전송한다.
⑪ Client는 전송된 T_7을 복호화 하여 R'_2를 얻고 Client가 보유한 R_2와 비교 검증을 통해 상호 인증 과정 완료한다.

아래 [그림 6]에서처럼 암호화하는 시점과 복호화 하는 시점은 모든 데이터가 SOAP이라는 방법으로 서로 전송하기 때문에 암호화한 각각의 데이터를 직렬화(Serialize)를 통해 구성한 다음 MTOM(Message Transmission Optimization Mechanism)하여 전송을 하며, 복호화 하는 시점은 역직렬화(Deserialize) 과정을 통해 데이터를 얻은 후 복호화 하게 된다.



[그림 6] 암호화/복호화 시점

다음은 세션키가 공유되었기 때문에 세션키로 대칭키 암호화를 한 데이터가 전송되는 방법이다[그림 7] 참조.



[그림 7] 메시지 처리

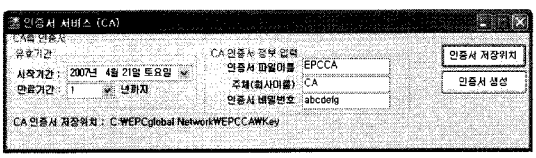
4. 제안 시스템 구현

본 논문에서는 닷넷 기반의 CA, 공인전자문서보관소 이용자(클라이언트) 및 공인전자문서보관소 업무 서버(서버)를 구축하고 해당 작업의 요청을 위해 Web Service 방법으로 데이터를 전송 및 전달받게 된다.

구현에 사용된 프로그램은 Microsoft Visual Studio 2005 C#, Microsoft Web Service Enhancements 3.0, Microsoft SQL Server 2005를 사용하였다.

4.1 CA의 인증서 생성 및 CA Web Service

다음 [그림 8]은 공인전자문서보관소 CA에서 개인키와 공개키, 그리고 자신의 키로 인증서를 생성하는 인터페이스이다.



[그림 8] CA 인터페이스

인증서를 생성하는 알고리즘은 다음과 같다.

```
// CA의 개인키와 공개키 생성
nRet = Rsa.MakeKeys(CAPublicKeyFileName,
CAPrivateKe yFileName, 1024,
Rsa.PublicExponent.Ex p_EQ_65537, 1024,
CACertPassword, Rsa.PbeOpti ons.Default, true);

//CA 인증서 생성
nRet = X509.MakeCertSelf(CACertFileName,
CAPrivateKey FileName, 1, CAExpireYear,
"C=KR;O=EPCGlobal Network; OU=RetailTech;CN="
+ CAName + "", "", X509.KeyUsage
Options.DigitalSignature |
X509.KeyUsageOptions.KeyCertSi gn,
CACertPassword, X509.Options.FormatPem);
```

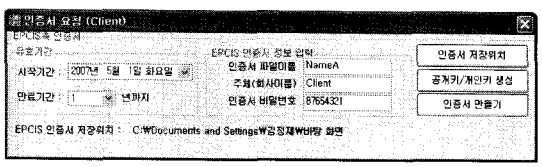
다음 알고리즘은 다른 원격지의 컴퓨터에서 공개키를 보내주면 인증서로 생성하여 전송시켜 주는 공인전자문서보관소 CA의 Web Service 알고리즘이다.

```
// CA의 개인키와 공개키 생성
nRet = Rsa.MakeKeys(CAPublicKeyFileName,
CAPrivateKe yFileName, 1024,
Rsa.PublicExponent.Ex p_EQ_65537, 1024,
CACertPassword, Rsa.PbeOpti ons.Default, true);

//CA 인증서 생성
nRet = X509.MakeCertSelf(CACertFileName,
CAPrivateKey FileName, 1, CAExpireYear,
"C=KR;O=EPCGlobal Network;
OU=RetailTech;CN=" + CAName + "", "",
X509.KeyUsage Options.DigitalSignature |
X509.KeyUsageOptions.KeyCertSi gn,
CACertPassword, X509.Options.FormatPem);
```

4.2 Client의 인터페이스 및 Web Service

다음 [그림 9]는 Client에서 개인키와 공개키를 생성하고 공개키를 공인전자문서보관소 인증서서버의 웹서비스에 요청하게 되면 인증서 서버에서는 인증서와 개인키를 가지고 전자서명한 후, Client 인증서를 생성하여 다시 Client로 전송해주게 된다.



[그림 9] Client의 인증서 요청 인터페이스

이때 CA에서 발급된 인증서를 클라이언트에게 전송할 때는 MTOM 메시지로 전송하게 된다.

다음 [그림 10]은 공인전자문서보관소 인증서 서버에서 Client로 전송하는 SOAP 메시지의 일부분이다.

```
- <soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope" soap:encodingStyle="http://
  - <soap:Body>
    - <CertMakeFromPublicKeyResponse xmlns="http://tempuri.org/">
      - <getFileResponse xmlns="http://localhost/WebService/CA">
          <fileName>Client.Cer</fileName>
          <fileData>mRmTUzcHjOBLOCZ0RKbvuWaVfukUaiqsnNAzGlrxja6JaK4wOCm4NLcqRTgOj0MumMr
        </getFileResponse>
      </CertMakeFromPublicKeyResponse>
    </soap:Body>
  </soap:Envelope>
```

[그림 10] 인증서 요청 SOAP 메시지

4.3 Server Web Service

다음 [그림 11]은 Server에서 Client로 메시지를 전송해 주는 SOAP 메시지이다.

<fileData>의 엘리먼트는 본 논문의 3.2 Message 암호화 전송과정에서의 랜덤값 R1을 클라이언트의 공개키로 암호화 한 후, 서버의 개인키로 전자서명 한 값으로, T1에 해당하는 메시지이며, 전송 단계시 Trace 디버깅 단계에서 캡쳐한 값이다.

```
- <soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope" soap:encodingStyle="http://
  - <soap:Body>
    - <T1_EncryptMessage xmlns="http://tempuri.org/">
        <SystemName>Client_127.0.0.1/4000</SystemName>
        <fileName>T1Msg</fileName>
        <T1_fileData>7Gw3LhJFven2JX1wgqyhO5KgOSIcG0S6NQ9Mu29NBXfkJYZIxzmfKkw4kvSbztiZDyN
      </T1_EncryptMessage>
    </soap:Body>
  </soap:Envelope>
```

[그림 11] T1 메시지 캡쳐

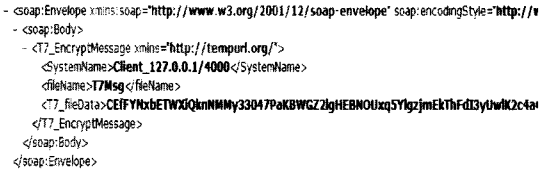
다음 [그림 12]은 Client에서 Server로 메시지를 전송해 주는 SOAP 메시지이다.

<fileData>의 엘리먼트는 랜덤값 R1을 획득하고 R2 메시지를 다시 보내주는 값으로 암호화 과정은 이전과 동일하다.

```
- <soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope" soap:encodingStyle="http://w
  - <soap:Body>
    - <T4_EncryptMessage xmlns="http://tempuri.org/">
        <fileName>T2Msg_127.0.0.1_4000</fileName>
        <T4_fileData>FtDmlV66LPd4yMj5C0tl2Al7cSPiJCiOnTHtvV8p8AR2AGnWK0BimxonB5XSi4By3rf1kBiz
      </T4_EncryptMessage>
    </soap:Body>
  </soap:Envelope>
```

[그림 12] T4 메시지 캡쳐

마지막으로 Server에서는 Session 값을 취득한 후, 클라이언트에 그 값으로 대칭키 암호화를 통해 그 값을 확인하는 과정이다.



[그림 13] T7 메시지 캡쳐

5. 실험 평가

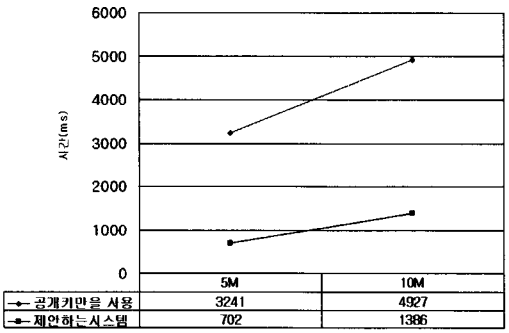
5.1 암복호화 시간

본 논문에서 사용되는 실험 결과 값은 5Mbyte, 10Mbyte XML 데이터를 가지고 실험평가 하였다. Request XML문서의 크기는 30Kbyte로 설정하였다.

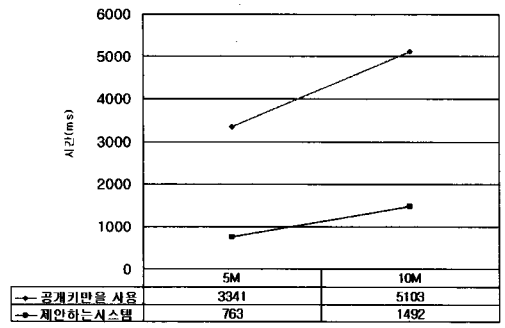
[표 2]와 [표 3]의 암복호화 시간은 은 클라이언트에서 요청하여 서버에서 응답을 받기까지의 시간을 그래프로 나탄 낸 것이다. 이 실험평가는 전체 동작 시간을 초점으로 맞추고 있다.

대칭키 알고리즘을 사용하지 않고 공개키 알고리즘만을 사용한 방법과 제안하는 시스템에서의 암복호화 시간을 비교하였다. 이때 CPU에 대한 점유율은 50%를 유지하도록 설정하였다.

[표 2] 데이터 암호화 시간



[표 3] 데이터 복호화 시간



데이터의 크기는 공개키만을 사용한 방법에서는 공개키만을 사용한 방법보다 약 4배 정도 빨랐으며, 이는 대칭키 알고리즘이 공개키 알고리즘보다 빠르기 때문이며, 데이터 전송되는 양도 제안하는 시스템이 약간 더 적음을 알 수 있었다.

하지만 제안하는 시스템의 경우 전송되는 데이터에는 약 2Kb 정도 증가하였는데 증가한 이유는 전자 서명에서의 해쉬값과 공개키로 암호화된 대칭키를 전송했기 때문이다.

5.2 안전성 평가

제안한 기법은 기존의 공개키 암호화 알고리즘만을 이용한 기법과 대칭키 암호화 알고리즘을 혼합한 시스템으로 전자문서는 모두 대칭키를 통해 암호화하여 사용자에게 전송시켜준다. 하지만 그에 따르는 보안상에 문제점의 발생을 최소화 하고, 기존의 공인전자문서보관소에서 제안하는 방법과 제안하는 시스템이 제공하는 전자문서에도 안전성 기능을 계속적으로 유지 할 수 있어야 한다. 제안한 기법에서는 시스템의 오버헤더를 줄이기 위해 대칭키로 암호화하고, 사용된 키만을 공개키로 암호화하여 전송하는 기능을 탑재하고 있다. 이에 수반되는 문제가 없는지 분석해 보고, 악의적인 사용자들에 의한 스니핑(Sniffing), 스푸핑(Spoofing), 재전송(Replay Attack)등의 공격들에 대해서도 안전한지 살펴본다.

5.2.1 사용자 인증

본 논문에서 제안한 시스템은 도메인에 공인전자문서를 등록 할 때, 인증을 위해 사용자의 개인키를 통해 전자서명한 데이터를 공인전자문서보관소의 공개키를 통해 암호화하여 데이터 전송을 한다. 여기서 인증서를 사용하는 이유는 불법적인 사용 및 유출을 막고, 완벽하게 사용자를 식별하기 위해서이다. 그리고 인증서를 사용한 공개키 암호화 방식은 현재까지 어떠한 공격에 대해서도 안전하며, 공개키와 개인키로 나누워져 있기 때문에 키 교환을 하는데 있어서도 편리하다.

5.2.2 스니핑 공격에 대한 안정성

사무실과 집, PDA등에서 사용하는 기기는 장치에 따라 암호화된 키를 분배하는 방식이 다양하다. 하지만 SOAP을 이용한 방식은 추가적으로, 암호화에 대한 프로세스 변경, 비즈니스 모델에 의한 변경 등 각각의 모듈별로 변경하기가 용이하고, 웹에 연결되어 있는 컴퓨터간에는 어떠한 데이터도 전송할 수 있는 장점이 있

다. 특히 UDDI 검색을 통해 서비스간의 WS 명서서만을 공개하고, 사용자는 그 명세서에 맞게끔만 데이터를 전해주면 모든 처리가 자동적으로 처리될수 있게끔 SOA(Service Oriented Architecture) 개념의 환경에 맞게끔 구성하고, 불필요한 서비스는 블랙박스 형태로 처리하여 구성할 수 있다. 제안하는 방식은 어떠한 방식으로 데이터가 전송되던 간에 데이터는 항상 암호화되어 전송되므로 불법적인 장치가 키를 가지고 있지 않는 한 디바이스 정보가 노출될 위험은 없다. 실제로 사용자 인증과 문서 암호화 과정 및 사용자 키 분배에 있어서, 대칭키의 비밀키는 난수 발생기에 의해 값에 의해 발생되고 상호간의 인증에 있어서는 세션이 이루어질 때마다 항상 다른 난수 R값을 적용하기 때문에, 키를 유추하는 것이 어렵다. 또한 인증서 교환으로 인해 공개키가 노출되더라도 복호화 하기위해서는 개인키 없이는 R값을 유추할 수 없기 때문에 안전하다.

5.2.3 스푸핑 및 재전송 공격에 대한 안전성

제안한 시스템의 암호화 방법은 스푸핑 및 재전송 공격으로부터 보호받기 위하여 클라이언트와 서버간의 난수값 R1과 R2를 이용하여 전자서명 및 전자서명 검증, 공개키 암호화, 개인키 복호화를 이용하여 불법적인 사용자의 인증 및 사용을 방지하고 있다. 3.2 절에서 기술한 사용자 인증 프로토콜에서 중요한 인증 정보인 난수 R1과 R2는 함께 연접하여 전자서명 및 암호화하여 상호교환하기 때문에, 중간에서 메시지를 가로채더라도 실제 메시지의 내용을 알 수 없을 뿐 아니라 인증이 이루어질 때마다 값이 매번 변하기 때문에 값을 유추하는 것은 불가능하다. 그러므로 실제 ID를 가지고 있지 않은 사용자가 중간에 메시지를 가로채어 재전송 하여도 인증 받을 수 없다.

6. 결론

본 논문에서는 공인전자문서보관소 업무 서버와 이용자 시스템 간에 전자문서 송수신시에 안전하게 메시지를 보낼 수 있도록 할 수 있는 암호화 시스템을 제안 및 구현하였다. 이는 ITU-T의 X.509 기본 원칙 을 그대로 지켰고, 「이용자시스템과 공인전자문서 보관소 간 연계인터페이스 기술 규격 V1.10」 중 송수신 인터페이스 규약에 부합하도록 노력하였으며, 메시지 복호화시 대량의 데이터를 속도 향상을 위해 대칭키를 사용하여 암호화 하였고, 여기에 사용된 키만을 개인키로 암호화 시켰으며, 데이터의 무결성을 위해 메시지를 전자서명을 통해 수정하는 것을 방지하였다. 또한 시스템간의 인증을 하기 위해 인증서를 통해 서로 인증할 수 있도록 하였다. 공인전자문서보관소와 이용자 시스템간의 안전한 송수신 서비스는 현재 기술 규격이 나와 있는 상태에서 일부 회사에 의해 상용솔루션이 부분적으로는 출시되어 있으나 아직은 미비한 실정이다. 향후 이에 대한 세밀한 기술연구 및 상용화를 포함한 검증을 통해 송수신 서비스의 질을 개선해 나가야 할 것으로 본다.

참고문헌

[1] 이만영, 김지홍, 류재철, 송유진, 염흥렬, 이임영, 전자상거래 보안기술, 생능출판사, 1999

[2] 최연희, 박미옥, 전문석, "디지틀 서명검증을 위임하기 위한 새로운 인증서 검증기법", 한국 인터넷 정보 학회 논문지, 제4권, 제 4호, pp53-64, 2003년 8월

[3] Bruce Schneier, Applied Cryptography, John Wiley & Sons, 1995.

[4] Eric A. Yong and Tim J. Hudson, OPENSSL toolkit, http://www.openssl.org.

[5] Naganand Doraswamy and Dan Hakins, The New Security Standard for the Internet, Intranets, and Virtual Private Networks, Prentice Hall PTR Internet Infrastructure Series, 1999.

[6] Loran M. KohnFelder, "Towards a Practical Public-Key Cryptosystem", B.S. Thesis, supervised by L. Adleman, MIT, May 1978.

[7] R. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems", Communications of the ACM, vol.21, no.2, pp.120-126, February 1978.

[8] ElGamal, T., "A Public Key Cryptosystems and a Signature Scheme Based on discrete Logarithms", IEEE Transaction on Information Theory, vol. IT-31, no. 4, pp. 469-472, July 1985.

[9] Denning. D. E., "Protecting Public Keys and Signature Keus", IEEE Computer, vol. 16, no. 2, pp. 27-35, February 1983.

[10] 김정재외 4명, "서명자의 신원정보 해쉬값을 이용한 실시간 인증서 상태 검증 메커니즘의 설계," 한국정보처리학회지 논문지C. Vol.13-C No. 02, 147~154pp., 2006. 04

[11] 한국무역정보통신-비씨큐어, "공인전자문서보관소 구축방안연구', 2003. 12.

[12] 한국전자거래진흥원, "공인전자문서보관소 산업환경분석", 2006. 1.

[13] 비씨큐어, "공인전자문서보관소 보안 및 핵심솔루션 소개", 2007. 10.
[14] 한국전자거래진흥원, "이용자시스템과 공인전자문서 보관소 간 연계인터페이스 기술 규격 V1.10", 2007년 8월
[15] 토피도, "공인전자문서보관소를 위한 차별화된 컨텐츠 보관전략", 2007. 10.

김 대 중(Kim Dae-Jung) [정회원]

- 2000년 8월 : 한국방송통신대학교 컴퓨터과학과 졸업(이학사)
- 2004년 2월 : 숭실대학교 산업기술정보대학원 정보통신공학과 졸업(공학석사)
- 2006년 2월 : 숭실대학교 일반대학원 컴퓨터학과 박사과정

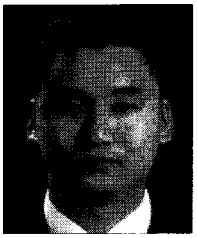
<관심분야>
보안, 정보통신, 암호학, RFID

김 정 재(Kim Jung-Jae) [정회원]

- 1999년 2월 : 영동대학교 컴퓨터공학과 졸업(공학사)
- 2001년 2월 : 숭실대학교 컴퓨터학과 졸업(공학석사)
- 2005년 8월 : 숭실대학교 컴퓨터학과 졸업(공학박사)
- 2006년 7월 ~ 현재 : (주)레일테크 기술연구소 수석연구원

<관심분야>
DRM, 암호학, RFID

이 승 민(Lee Seung Min) [정회원]



- 2004년 2월 : 한서대학교 컴퓨터정보학과 졸업(이학사)
- 2006년 2월 : 숭실대학교 일반대학원 컴퓨터학과 졸업(공학석사)
- 2007년 2월 ~ 현재 : 숭실대학교 일반대학원 컴퓨터학과 박사과정

<관심분야>
보안, 정보통신, RFID, 네트워크, PKI

전 문 석(Jun Moon-Seog) [정회원]

- 1981년 : 숭실대학교 전자계산학과(공학사)
- 1986년 : University of Maryland Computer Science(공학석사)
- 1989년 : University of Maryland Computer Science(공학박사)
- 1989년 3월 ~ 7월 : Morgan State University 조교수
- 1989년 ~ 1991년 : New Mexico State University Physical Science Lab 책임 연구원
- 1991년 ~ 현재 : 숭실대학교 부교수

<관심분야>
전자상거래 보안, 인터넷 보안, 멀티미디어 보안, 인증시스템, PKI, RFID