공개키 기반 구조에서 안전한 메일 전송을 위한 클라이언트 메일 보안 시스템 설계 및 구현

정창렬* · 고진광*

Design and Implementation of a Client Mail Security System for Secure Mail Exchange using Public Key Infrastructure

Chang-Ryul Jung* · Jin-Gwang Koh*

이 논문은 2002년도 순천대학교 공과대학 학술재단연구비에 의해 연구되었음

요 약

최근 인터넷을 기반으로 한 정보기술의 발전은 정보처리 및 정보교환이 활발해 짐에 따라 인터넷 이용의 많은 부분이 전자화된 문서를 인터넷 메일을 통해 주고받고 있다. 이는 오픈 네트웍을 통해 전자문서를 보내고 송, 수신함에 있어 문서정보에 대한 안정성이 위협 받고 있다. 특히 중요문서를 주고받을 때는 그 위협 정도는 매우 높다. 따라서 본 연구에서는 이러한 전자 문서들을 안전하게 전송할 수 있도록 하는 클라이언트 메일 보안 시스템을 설계 구현하였다. 그러므로 전자문서 정보를 인터넷을 통해 전달할 때 발생 가능한 정보의 조작이나 변질, 도용으로부터 중요 문서정보를 보호하도록 공개키 기반 구조에서 안전한 전송을 위한 암호화된 메일 전송과 배달증명 그리고 메일메시지 무결성을 보장 하도록 한다. 뿐만 아니라 윈도우즈용 GUI 인터페이스 환경에서 공개키를 기반으로 한 SET프로토콜을 이용하여 전문적인 지식이 없는 일반 사용자도 쉽게 사용할 수 있는 사용자 인터페이스와 공개키 암호화 알고리즘을 적용한 메일보안시스템을 개발하였다.

ABSTRACT

Recently, the Internet enhanced by developement of IT makes the processing and exchanging of information, As the Internet is sending and receiving digitized documents over the Internet e-mail system. The security of document information is being threated when exchanging digitized documents over an open network such as the Internet. The degree of threat is even higher when sensitive documents are involved Therefore, in this paper, the secure e-mail system on a client is designed and implemented in order to make secure exchanging of digitized documents. By using the public key infrastructure in which encrypted mail transmission, proof of delivery and integrity of the message are garanteed, unauthorized manipulation, illegal acquisition and mutual authentication problem can be prevented in order to secure the document information which is crucial and sensible when exchanging the digitized document over the Internet. Futhermore, by using the SET protocol based on public key cryptography, the secure mail system is designed and implemented in order for the users not having any professional knowledge to deal with the system easily and friendly in GUI environment..

키워드
공개키, 메일시스템, SET프로토콜, 공개키 기반구조

*순천대학교 컴퓨터과학과

접수일자 : 2002. 11. 25

Ⅰ. 서 론

디지털 정보혁명의 주도적인 패러다임은 인터넷을 기반으로 한 시공간의 초월한 의사소통을 가능하게 하고 있다. 이러한 의사소통의 중추적인 역할을 하는 것은 전자메일을 통한 메시지를 주고 받는 것이다. 그러나 인터넷을 통한 중요문서를 주고받을 때에는 안전성에 대한 위협을 받아 메시지 전달의 위험 부담이 높다[3]. 그러므로 전자문서를 인터넷을 통하여 전송될 때 항상 안전하고 신뢰할 수 있도록 메시지의 무결성(Integrity)이 보장되어야 한다. 때문에 최근 메일 보안에 대한 많은 관심과 논의가 되고 있다. 이러한 논의는 문서의 암호화, 인증, 부인방지, 전자서명 등 주요 보안 기술을 이용하는 것이다.

따라서 본 연구는 개인의 메시지 정보의 암호화와 공개키 방식의 암호화 시스템을 이용하여 데이터 전송중에도 메시지에 대한 위협으로부터 보호한다. 뿐만 아니라 프로토콜을 새롭게 변경하지 않는 한 변경이 불가능하게 하기 위해 SET 프로토콜을 이용한다. SET 프로토콜은 RSA 공개키 알고리즘, DES 대칭키 알고리즘, 메시지 축약 등의 알고리즘들을 사용함으로써 메시지의 무결성, 기밀성(secrecy)이 가능하도록 한다. 본 연구의 시스템은 클라이언트 상에서 자바를 기반으로 한 메일보안시스템이다.

본 연구의 제1장은 연구의 필요성과 목적을 기술하고, 제2장에서는 공개키 기반구조에 대해 고찰한다. 제3장에서는 본 연구에서 사용된 SET프로토콜의 개념과 메일보안시스템의 설계하고, 제4장에서는 설계된 모듈들을 구현하여 기존의 메일시스템과 비교분석한다. 마지막 제5장에서는 결론 및 향후 연구에 대해 기술한다.

Ⅱ. 공개키 기반 구조의 구성요소 및 요구사항

공개키 기반구조는 인증서의 발급, 사용 및 취소와 관련된 서비스를 제공하며, 공개키 기반구조의 환경을 구축하는 주요 객체는 인증기관(Certification Authority), 인증서 저장소(Certificate Repository) 그리고 서비스 제공 주체로 구분된다[12].

그림 1은 공개키 기반구조의 구성객체의 기본적인 트랜젝션을 보여준다.

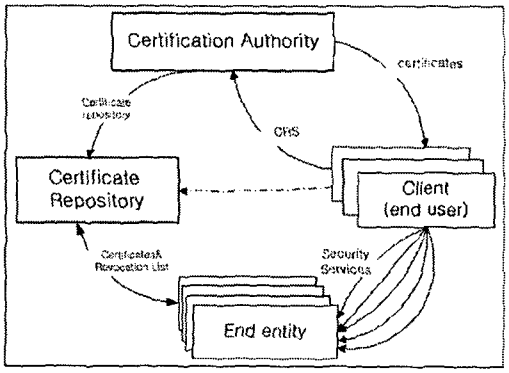


그림 5. 공개키 기반구조의 구성요소
Fig. 3 composition of PKI

Certification Authority(CA) : 공개키 기반구조를 구성하는 가장 핵심적인 주체로 그 역할 및 기능에 따라 계층적으로 구성하며 여러 명칭에 따라 몇 가지로 구분된다. 공개키 기반구조의 전반에 사용되는 정책과 절차를 수행하여 수립하는 Policy Approving Authority(PAA)와 PAA아래 계층에서 자신의 도메인 내의 사용자와 CA가 따라야 할 정책을 수립하는 Policy Certification Authority(PCA)와 일반적으로 인증기관의 기능을 수행하는 CA로 구분되어진다. 실제로 CA에서는 사용자의 공개키 인증서를 발행하고 또 필요에 따라 취소하고, 사용자에 공개키를 전달하고, 등록기관의 요청에 의해 인증서를 발행하고 되돌려주는 기능을 한다. 또한 인증서 소유자에 대한 정보관리, 디렉토리 서버관리, 인증서 취소목록(CRL)관리 등을 한다[7-9]

CA의 기능 중 발행된 인증서 및 CRL을 실시간으로 인증서 저장소에 공고할 수 있는 기능이 요구되며, 인증서 발행은 사용자가 직접 공개키를 생성하여 사용자의 정보와 해당 공개키에 대한 인증을 요구하는 SPKAC형태 및 PKCS#10형태의 인증서를 모두 발급 가능해야 한다. CA는 항상 CA인증서와 상호 인증서 쌍과 CRL을 공개 할 수 있어야 한다.

Certificate Repository(CR): 디렉토리 서비스라고 하며 인증서와 인증경로를 찾기 위해서 사용된다. CA에 발행된 공개키 인증서와 CRL을 저장하고 열람할 수 있도록 검색기능을 제공해야 한다. 보통 X.509표준에 따른 경우 DAP(Directory Access Protocol)이나 LDAP(Lightweight DAP)을 제공한다[9]. 인증서와 상

호인증서 쌍은 유효기간이 경과된 후에도 서명의 검증을 위하여 일정기간동안 디렉토리에 저장 된다[6].

Client, end user: 공개키 기반구조 내의 사용자는 사람뿐 만 아니라 사람이 이용하는 스텝 모두를 의미한다. 이들 기능은 자신의 비밀키/공개키 쌍을 생성하며, 공개키 인증서를 요청한다. 전자서명서를 검증하며 그 상태를 결정한다.

Certificate Revocation List는 예정된 유효기간의 만기일이 도래하기 전에 취소된 인증서에 대한 정보 취소목록이다[10].

Certificate는 공개키을 인증하고 보증하는데 이용되는 서명확인과 같은 것으로 서로에 신뢰하고 사용할 수 있도록 사실을 확인하는 것이다. 이러한 것 이외에도 관련하는 기술이 많이 연결되어 기반구조를 이루고 있다.

본 연구에서는 이러한 기반 구조에서 현재 전자상거래에서 많은 검증과 사용이 이루어지고 있는 SET 프로토콜을 이용하여 공개키 알고리즘을 적용한다.

III. SET 프로토콜과 암호메일시스템

Visa카드사와 Master 카드사를 주축으로 하여 어떤 형태의 네트워크에서도 안전하게 신용카드를 사용할 수 있도록 하기 위해 SET(Secure Electronic Transaction)프로토콜을 공동으로 개발하였다. 네트워크 상에서 안전하게 신용카드를 사용하기 위해서 여러 가지 보안요소를 만족해야 하는데 SET프로토콜은 안전하고, 기밀성을 보장하고 인증된 수신자만 볼 수 있도록 암호화한다. 또한 제 3자에 의해 메시지가 변경되지 못하도록 데이터 통합성을 보장하는 프로토콜이다[13].

3.1 SET 프로토콜의 암호처리

두 송수신자간에 전송되고 수신하는데 서로 연결하여 사용한다.

3.1.1 송신자 측면의 알고리즘

① 해쉬 알고리즘을 이용하여 메시지 축약을 한다. 송신자의 비밀키를 이용해서 암호화된 전자서명을 생성한다. ② 송신자에 의해 만들어진 대칭키로 암호화 한다 ③ 대칭키는 수신자의 공개키를 가지고 암호화하여 전자봉투를 생성 한다 ④ 전자봉투와 암호화된 메시지를 메시지 수신자에게 보낸다.

3.1.2 수신자 측면의 알고리즘

① 수신한 전자봉투를 수신자의 비밀키로 이용하여 복호화 한 후 송신자에 의해 만들어진 대칭키를 생성 한다 ② 대칭키를 가지고 암호화된 메시지를 복구하여 송신자의 암호문을 만들어낸다. ③ 송신자로부터 전달된 암호문과 송신자의 전자서명을 복호화한 메시지를 비교하여 같다면 전송도중 에러가 발생하지 않은 것으로 판단하여 정상적으로 메시지를 수신한다.

3.2 암호 시스템

암호시스템은 일반적으로 암호화되지 않은 상태의 원문(plain text)을 암호문(cipher text)으로 만드는 암호화(encryption) 과정, 역으로 암호문을 원문으로 변화시키는 복호화(decryption) 과정, 그리고 이 과정에 사용되는 암호화 키(cryptographic key)와 그 관리 등이 정보보호를 위한 일련의 과정들을 암호시스템이라 한다.

3.2.1 대칭키 암호화시스템

하나의 키를 이용하여 주어진 데이터를 암호화하고 동일한 키를 이용하여 암호문을 해독하여 원래의 데이터를 생성한다. 이때의 암호 알고리즘에 사용되는 키를 대칭키 또는 비밀키라고 한다. 비밀키는 상대적인 공개키의 크기보다 작아서 적은 자원에서 효과적인 암호시스템을 구축할 수 있어 속도가 공개키 암호화시스템 보다 현격히 빠르다. 뿐만 아니라 알고리즘의 내부구조가 간단한 치환과 순열의 조합으로 되어 있어 시스템 환경에 맞는 적절한 암호 알고리즘을 쉽게 구현할 수 있다[2].

가장 대표적인 예로서는 DES가 있으며 DES를 변형한 GOST, IDEA, FEAL, RC2와 RC4, 3DES 그리고 1993년 미 국립 보안국에서 컴퓨터 칩용으로 개발한 SKIPJACK 등이 있다.

3.2.2 공개키 암호화 시스템

서로 상이한 한 쌍의 키에 의해 작동되는 암호화

함수 E와 복호화 함수 D로 설계하여 종래 암호화 시스템에 있어서 키 교환 문제점을 해결하였다. 공개키 방식(Public Key Method)은 키 쌍(key pair)을 생성한 후 하나의 키를 상대편에게 공개한 후 이를 이용하여 데이터를 보호하거나 무결성, 인증, 부인봉쇄 등을 만족시킨다[1].

이러한 특징의 공개키 방식은 두 가지 용도로 이용되는데 하나는 데이터를 보호하는데 이용하는 것과 다른 하나는 데이터를 보낸 사실에 대해 부인봉쇄에 이용된다.[8].

3.3 암호 메일시스템 설계

3.3.1 메일보안시스템 설계

① 공개키 메일보안 시스템 설계

공개키 암호 알고리즘을 이용하여 데이터를 암호화하는 경우 상대방의 공개키를 이용하여 암호화할 때와 자신의 비밀키로 데이터를 암호화하여 전자 서명한다[7]. 그림 2는 공개키 암호화 알고리즘 설계 모듈을 나타내고 있다.

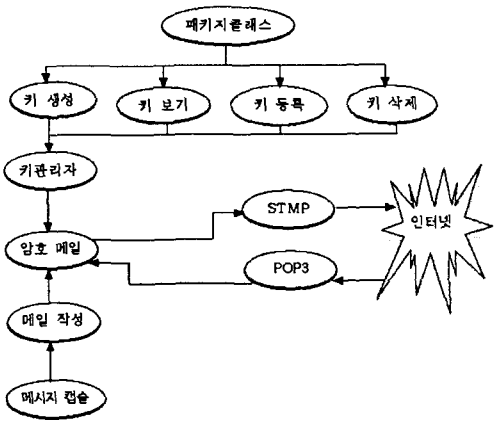


그림 2. 공개키 암호화 알고리즘 설계 모듈
Fig. 2 Design module of public key Cipher algorithm

이러한 암호시스템은 암호화와 복호화의 과정에서 사용되는 키의 사용에 따라 그림 3과 같은 범주로 나눈다.

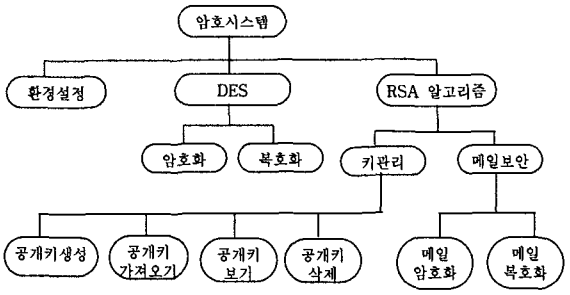


그림 3.메일보안시스템 전체구조
Fig. 3 Architecture of mail security system

② 배달증명

배달증명은 부인방지 서비스의 일환으로 컴퓨터통신망을 통해서 주고받는 전자문서에 대한 올바른 수신이 되었는가를 증명해 주는 서비스이다. 이러한 서비스를 전자메일에 적용함으로써 현행 등기우편과 같은 서비스가 진행되면서 중간에 조정자가 있어서 수신자와 발신자의 사이에서 이러한 서비스를 제공하는 방식이다. 중간에 조정자를 두어 이용하는 방식은 사용자의 증가로 프로토콜의 부담이 높아져 사용자의 부담이 가중되는 문제가 있을 수 있다. 본 연구에서는 SET프로토콜의 기반에서 조정자를 이용한 Zhou and Gollmann의 방식을 이용하여 조정자가 분배를 공평하게 하여 배달증명을 하도록 한다.

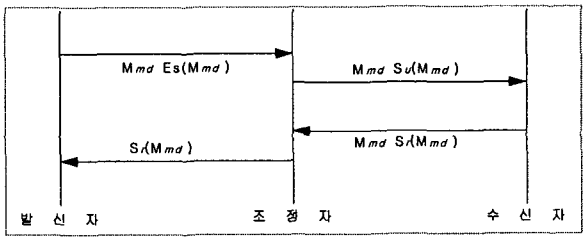


그림 4. 암호메일의 배달증명 프로토콜
Fig. 4 delivery-proof protocol of cipher mail

그림 4는 배달증명을 나타내는 프로토콜이다. 발신자의 메시지를 해쉬 알고리즘에 의해서 암호화 하여 다시 수신자의 공개키로 암호화하여 메시지를 조정자에게 전송한다. 전송된 메시지는 다시 조정자에 의해서 배분되어 수신자에서 보내어 진다. 보내어진 암호화된 메시지가 수신자에게 전달되면 수신증명이 된

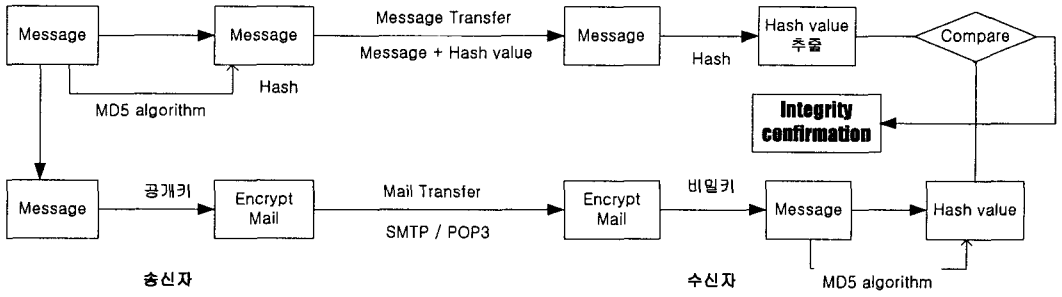


그림 6. 메일 메시지의 무결성 검증 모듈
Fig. 6 Integrity of mail massage

다. 이것을 다시 수신자에 의해서 수신메시지에 대해 다시 RSA공개키 알고리즘에 의해서 암호화하여 조정자를 거쳐 수신자에게 전송되어 비로소 안전하게 배달이 이루어 졌다고 확인 할 수 있어 암호화의 기본 기능인 부인방지를 할 수 있는 배달증명이 된다.

③ 메일보안 시스템의 클래스 모듈

본 연구의 메일보안시스템은 클라이언트에서 제공된다. 그리고 메시지의 암호화와 복호화 및 배달증명을 담당하는 메일보안클래스와 메일 메시지를 저장하고 관리하는 메시지 클래스를 핵심으로 하고 있다.

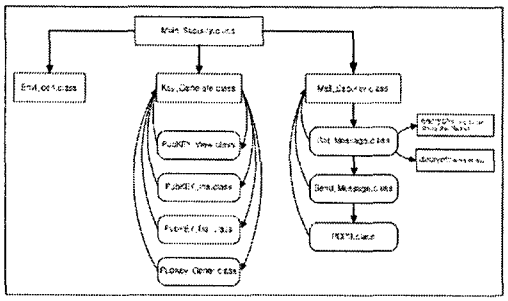


그림 5. 메일 보안 시스템의 전체 클래스 모듈
Fig. 5 Total class module of mail security system

메일 보안 시스템은 메일 클라이언트를 기반으로 하기 때문에 정보보호 서비스가 가능하며 메일의 메시지를 암호화 및 서명하여 전송함으로서 메시지가 전송되는 네트워크에서 도청되거나 불법 변조되는 등의 보안상의 문제에 효율적으로 대처할 수 있게 되었다.

그림 5는 메일보안시스템의 전체적인 클래스 모듈도이다. 각각의 클래스들은 ①Main_Security.class: 메인 메일 클래스를 제공한다. ②Envi.con.class : 환경설정에 필요한 데이터를 입력받아 preferences파일로 저장. ③Key_Generate.class: 생성된 키를 KeyManager 인스턴스가 키를 파일로 저장, 공개키를 생성. ④PubKEY_view.class : 키 관리자 파일에서 선택한 공개키를 등록하며 이름을 리스트 박스에 나타낸다. ⑤PubKEY_Ins.class : 키 관리자 파일에 공개키를 등록하고, ⑥PubKEY_Del.class : 키 관리자 파일에 공개키를 삭제한다. ⑦PubKey_Gener.class : RSA공개키 알고리즘에 의해 키를 생성한다. 이때 생성된 키는 KeyManager에 의해서 관리된다. ⑧Mail_Encrypt.class : 암호시스템의 핵심적인 클래스로 메시지를 실질적으로 암호화, 복호화 하면서 전자 서명도 확인한다. ⑨SendMessage.class : 새로운 메시지를 작성하고 메일전송프로토콜에게 메시지와 수신자 공개키를 전달한다. ⑩Get_Message.class : 헤더와 본문으로 구성된 E-Mail 메시지를 캡슐화 한다. ⑪ SMTP.class : E-Mail을 보내기 위해 SMTP 서버와의 연결을 관리 한다. ⑫POP3.class : E-Mail을 수신하기 위해 POP3 서버와의 연결 관리 한다.

④ Mail_Encrypt 클래스 설계

Mail_Encrypt.class 클래스는 크게 네 개의 메소드 부분으로 구성되는데 각 구성요소들은 아래와 같은 역할을 한다.

getMessages()

초기화 파일(preferences)에서 필요한 정보를 추출하여 POP3 클래스를 사용하여 POP3 서버에 연결하여 메시지를 가져온다. 또한 메소드 select Messaged(int index)에서 decrypt(String body)를 가져온 메시지를 복호화하여 메시지 영역에 출력한다.

sendMessage(Message, String remoteName)

초기화 파일(preferences)에서 필요한 정보를 추출한 다음 encrypt (String body, String their Name) 메소드를 호출하여 메시지를 수신자의 공개키로 메시지를 암호한 후 SMTP 서버에 연결하여 암호화된 메시지를 전송한다.

그림 6은 메일 메시지의 무결성의 검증 모듈을 나타내는 것으로 송신자로부터 송신한 메시지나 그 메시지를 해쉬 알고리즘인 MD5알고리즘에 의해 해쉬된 데이터를 함께 보낸다. 그러면 송신자는 발신자로부터의 데이터에 대해 다시 해쉬 알고리즘을 이용 원래의 해쉬를 추출하여 발신자에게 보낸 해쉬값과 비교한다. 만일 같으면 내용이 바뀌지 않았다는 메시지의 무결성을 확인하게 된다.

encrypt(String body, String theirName)

실제적인 암호화 작업을 수행하는 메소드로 다음의 순서로 진행되어 암호화를 한다. ① 키 관리자로부터 송신자 이름, 송신자 비밀키 그리고 수신자의 이름으로 수신자 공개키를 획득한다. ② 본문 암호화를 위한 세션키(DES 알고리즘)을 생성한다. ③ 작성된 메시지 본문을 ②에서 생성한 세션키로 암호화한다. ④ ②에서 생성된 세션키를 수신자의 공개키(RSA 알고리즘)로 암호화 한다. ⑤ 다음은 메시지 서명을 한다. 메시지서명하는 부분은 암호화 안된 원래의 메시지를 해쉬함 수 MD5로 메시지를 축약하고 축약된 메시지를 수신자의 공개키(RSA 알고리즘)로 암호화한다.

⑥ 전 단계에서 생성된 송신자 이름, 암호화된 세션키, 서명된 메시지, 암호화된 메시지를 출력 스트림으로 연결한 후 전체 데이터를 바이트 배열로 변환한다. ⑦ 암호화된 메시지를 인식할 수 있도록 base64 스트링 앞에 "CipherMail:"이라는 데이터를 추가한 후 base64로 인코딩한 긴 스 트링을 40문자마다 "/r/n"을 삽입하여 새로운 행에 삽입한다. ⑧ ⑦에서 생성된 base64 스트링을 리턴한다.

decrypt(String body)

암호화된 메시지를 복호화 하는 메소드로 encrypt() 메소드의 역순으로 진행된다. 진행 과정은 다음의 순서로 이루어진다. ① 수신된 메시지가 "CipherMail:"로 시작되지 않으면 암호화되지 않은 메시지임으로 복호화를 하지 않고 바로 리턴 한다. ② 수신된 메시지는 40문자마다 줄을 바꿈으로 암호화 과정에서 삽입한 "/r/n"을 제거한 스트링을 생성한다. ③ ②에서 생성된 스트링을 base64로 디코딩하여 바이트 배열로 변환 한다. ④ 생성된 바이트 배열로부터 송신자의 이름, 암호화된 세션키, 전자서명, 암호화된 메시지를 순서대로 읽는다. ⑤ 키 관리자로부터 비밀키를 획득하여 세션키를 복호화 한다. ⑥ 복호화된 세션키로 암호화된 메시지를 복호화 한다. ⑦ 송신자의 공개키를 키 관리자에서 획득하여 서명을 검증한다. 만일 송신자 공개키를 획득 못하면 서명 검증을 하지 못하게 된다. 송신자의 공개키를 획득하면 메시지를 검증 한다. ⑧ 복호화 된 메시지를 리턴 한다.

IV. 암호 메일시스템 구현

4.1 구현 환경

연구의 시스템 구현 환경은 운영체제는 Windows 98을 사용하였으며 설계 구현 도구로는 JAVA JDK1.3, JAVA Pad, IAIK JCE Library[5]를 연결하여 사용하였다. 메일보안시스템을 구현하기 위해 통신 프로토콜을 이용한다. 메일 송신을 위한 POP3 프로토콜과 메일 수신을 위해 SMPT프로토콜이다.

4.2 메일보안시스템 환경 설정

메일을 송수신하기 위해서 통신프로토콜이나 공개키 생성을 위한 키 매니저나 사용자의 E-mail 주소 등을 설정해야 한다. 환경설정은 메일보안 시스템의 풀다운 메뉴의 환경설정에서 바로 할 수도 있다. 최소한 한번 이상의 환경설정이 이루어진 상태라면 환경설정값이 파일로 저장되어 있어 재설정 할 필요

없다.

저장된 파일명은 "preferences" 라는 파일로 저장된다. 환경을 수정하여 재설정하고자 한다면 저장된 파일에서 바로 수정이 가능하며, 파일이 수정되면 바로 새로운 값으로 리플레이스 된다.

이때 저장되어진 파일명은 "preferences"로 원래의 파일에 자료만 오버라이트 되어 저장 된다.

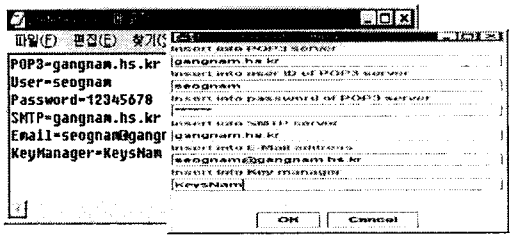


그림 7. 메일보안시스템의 환경설정 및 값
Fig. 7 environment setting and value of mail security system

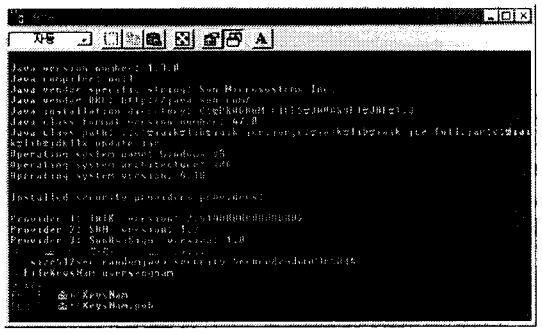
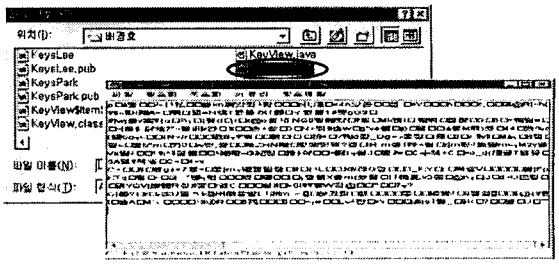


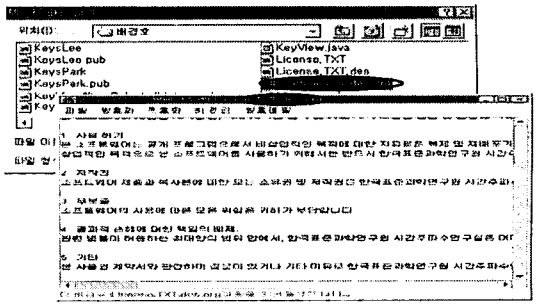
그림 8. 키 생성 결과
Fig. 8 Key generator result

4.3 문서 암호화 및 복호화

일반데이터 문서를 암호화 하여 저장하여 보관 하고자 경우 개인의 비밀키를 이용하여 암호화 하여 자료를 보관할 수 있다. 뿐만 아니라 일반 문서를 암호화해서 저장함으로써 자료를 효율적으로 관리가 가능하다. 그리고 일반문서나 자료를 암호화해서 수신자에게 보낼 수 있는데 보낼 때 SET프로토콜을 이용하기 때문에 안정성과 메일 메시지의 무결성을 보장받을 수 있다. 일반 문서를 암호화하기 위해서는 메일보안시스템의 메뉴에서 암호화를 이용하여 암호화하고 복호화 하고자 할 때복호화 메뉴를 이용하여 자료를 복호화 할 수 있다.



암호화된 파일을 다시 복호화 하면 .org라는 확장자 붙어(.des.org)서 원래의 문서로 복호화됨을 의미하고 복호화된 파일은 원문서와 구분된다.



암호화에 사용되는 키는 환경설정에서 키 생성기를 통해서 키를 생성한다. 생성된 키는 자동으로 DOS의 텍스트 창에 확인 할 수 있다. 그림 7, 8은 키를 생성하여 보여주는 결과이며, 암호화와 복호화는 생성된 키에 의해 실행된다.

4.4 메일보안시스템을 통한 암호 메일 전송

생성된 공개키를 이용하여 메일보안 시스템을 통해서 보내고 수신한 메일을 읽을 수 있는데 수신된 메일을 읽을 때는 수신자의 비밀키가 있어야만 해독하여 읽는다. 만약 수신자가 비밀키가 없다면 정상적인 메일전송이 되었음에도 수신된 메일메시지를 읽지 못한다.

그림 10은 메일보안 시스템을 통해서 메일을 수신한 경우와 수신자의 비밀키가 없는 일반 outlook Express에서 메일을 수신한 예를 보여주고 있다.

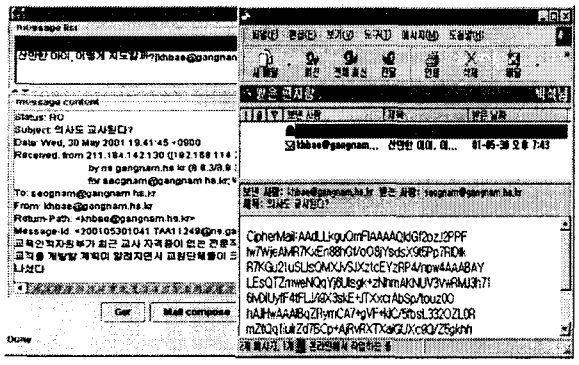


그림 10. 메일보안시스템에서 메일 읽기 와 일반 outlook에서 메일 읽기 비교
Fig. 10 compared mail reading and general mail in mail security system

outlook Express에서는 메일 내용을 알아 볼 수 없게 특수 문자로 표현되어 있는 것은 수신자가 메일을 통신프로토콜을 통행 메일을 수신하거나 아니면 통신프로토콜을 이용해서 해킹을 하여 메일이 정상적으로 수신이 되어도 원래의 수신자의 비밀키가 없으면 수신된 메일 내용을 읽을 수 가 없다. 이는 정상적으로 수신이 되어 메일 메시지의 무결성이 보장된다. 메일보안시스템을 이용해서 수신된 메일 메시지를 수신자의 별다른 동작이나 작업 없이 읽고자 하는 메일을 선택하고 Get버튼을 클릭하면 바로 메일메시지를 읽을 수 있다. 또한 본 연구의 메일 보안 시스템은 SET프로토콜을 기반으로 하고 있기 때문에 다른 어떤 시스템보다 안전한 메일 송수신이 가능하다.

그리고 메일을 상대방에게 보내고자 할 경우 상대방의 공개키를 알고 있어야 하는데 상대방의 공개키는 미리 등록을 해두어서 상대방의 공개키 읽어올 수 있도록 하여 원하는 수신자에게 손쉽게 메일을 전송할 수 있다. 그리고 메일 메시지를 보내어지는 동시에 암호화가 되어 전송이 된다. 그러므로 사용자는 전문적인 지식이 없이도 쉽고 편리하게 메일을 안전하게 암호화하여 무결성과 기밀성, 그리고 배달증명을 보장받을 수 있다.

4.5 비교 분석

본 연구에서 구현한 메일보안시스템은 일반적으로 이용되고 있는 PGP나 PEM와 비교 분석한다. PGP나 PEM는 MIME프로토콜을 이용하여 전자메일을 여러 알고리즘에 적용하여 메일 보안에 많이 사용되는 메커니즘이다. 이러한 PGP와 PEM의 보안성에 대해 비교 분석한다. 분석된 결과는 표 1과 같다.

표 1. 메일시스템의 보안 기능 비교분석
Table. 1 compared analysis of mail system

	PGP	PEM	구현된 메일보안시스템
메시지 무결성	○	○	○
메시지 기밀성	○	○	○
부인방지	○	○	○
메시지 인증	○	○	○
메일 배달증명	×	×	○
SET프로토콜	×	×	○
공개키기반구조	×	×	○

표 1은 PGP와 PEM에서 제공하지 않는 메일 배달증명과 공개키 기반구조 와 SET프로토콜을 이용하여 안전한 전송할 수 있는 메일보안 시스템을 비교 분석되는 것을 보여주고 있다.

V. 결 론

문서정보 보안을 위해 최근 전자 문서의 암호화, 인증, 부인방지, 전자서명 등의 개인의 전자 정보에 대한 암호화에 대한 관심이 고조되고 있다. 특히 일상 사용자들이 가장 많이 이용하는 전자 우편은 메일 메시지에 대한 정보보호가 꼭 필요하다.

따라서 본 연구에서는 이러한 정보보호를 위해 이식성과 호환성이 뛰어난 JAVA을 이용하여 메일보안시스템을 설계 및 구현 하였다. 특히 구현된 메일보안 시스템은 전자 문서들을 공개키 기반구조에 암호화하여 전달함으로써 정보보안 기능을 향상시켰다. 또한 정보를 인터넷을 통해서 전달할 때 발생 가능한 정보의 조작, 정보의 불법적인 획득 등으로부터 송 수신자간의 전송되는 중요한 문서정보의 신뢰성

을 높이기 위해 메일 메시지의 무결성을 보장하도록 하였다. 뿐만 아니라 윈도우즈용 인터페이스 GUI 환경에서 SET프로토콜과 공개키 암호화 알고리즘을 이용하여 전문적인 지식이 없는 일반 사용자도 쉽게 사용할 수 있는 메일보안시스템을 구현하였다.

향후 연구로는 연구범위를 확대하여 메일에 국한되지 않고 전자문서가 이용되고 있는 통합된 환경에서 정보보호가 이루어질 수 있도록 하는 연구가 필요하다.

참고문헌

[1] Jonathan K., 「JAVA Cryptography」, O'Reilly & associate,Inc, 1998.

[2] Matsui, M., "Linear Cryptanalysis Method DES cipher", Advanced in crtpyology-EUROCRYPT'93, LNCS 765, 1994

[3] William S., 「S/MIME : E-Mail Gets secure」, 1998.

[4] Jonathan B. Postel. "SIMPLE MAIL TRANSFER PROTOCOL", http://www.cis.ohiostate.edu/rfc/rfc0821.txt.

[5] IAIK Library JCE. http://jcewww.iaik.tu-graz.ac.at/

[6] M. Myers, C. Adams, D. Solo, and D. Kemp, Rfc 2511, "Internet X.509 Certificate Request Message Format", IETF X.509 PKI(PKIIX) Working Group., March, 1999.

[7] R. L. Rivest, A. Shamir, and L. Adlman, "A *Method for Obtaining Digital Signatures and* Public-key Cryptosystems", ACM, Vol.21, No. 2, pp.644-654, Feb. 1987.

[8] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme based on Discrete Algorithm", IEEE Trans. Information Thory, VIT-22, NO.6, Nov. pp.654-664, 1976.

[9] Matt B, Joan F, and Jack L. "Decentralized trust management. In Proc. 1996 IEEE Symposium on Security and Privacy, 1996.

[10] N. A. Nazario, "CA-browsing System-A Supporting Application for Global Security Services", ISOC Symposium on Network and Distributed System Security, SanDiego, pp.123-128, Feb. 1994.

[11] J. G. Koh, T. H. Kim, C. R. Jung, et., "Implementation of Cipher-mail System using SET Protocol in Clients", Proceeding of the International Conference on Parallel and Distributed Proc. Techniques and Application, PDPTA'2001, Las vegas, Nevada, 25-26, June, 2001.

[12] 반용호, 홍주형, 김종훈, "PKI기반 전자거래를 위한 공개키 인증시스템 설계 및 구현", 동아대학교 부설 정보기술연구소 논문집 Vol.8, No.1, pp.131-139, 2000.

[13] 이만영외, 「전자상거래 보안기술」,생능출판사, 2000.

저자소개

정창렬(Chang-Ryul Jung)

1995년 광주대학교 전자계산학과 졸업(학사)

1999년 순천대학교 교육대학원 컴퓨터교육학과 졸업(석사)

2002년 순천대학교 대학원 컴퓨터과학과 (박사수료)

※관심분야: Information Security, Image Processing, Mobile Agent, Authentication

고진광(Jin-Gwang Koh)

1982년 홍익대학교 컴퓨터공학과 졸업(학사)

1984년 홍익대학교 대학원 컴퓨터공학과 졸업(석사)

1997년 홍익대학교 대학원 컴퓨터공학과 졸업(박사)

1984. 3.~1998. 2. 송원전문대학 전자계산과 전임 강사

1988. 3.~현재 순천대학교공과대학 정보통신공학부 정교수

2001. 3.~2002. 9. 순천대학교 정보전산원장

※관심분야: Database, Information Security, Electronic Commerce