자바 기반의 배달증명이 가능한 전자메일 시스템 구현

우 준[†] · 하 영 국[††] · 임 신 영[†††] · 이 재 광[††††]

요 약

전자메일은 인터넷에서 개인간의 문서 및 서신 등의 정보뿐만 아니라 기업간의 공식적인 문서 교환에 이르기까지 다양한 용도로 이용되고 있다. 하지만 데이터에 대한 보안은 기대에 미치지 못하고 있는 실정이기 때문에 정보보호 서비스를 제공할 수 있는 전자 메일 시스템은 대단히 중요하다고 할 수 있다. 본 논문에서는 기본적인 정보보호 서비스 외에 기존의 메일 시스템에서는 제공되지 않으며, 발신자의 문서가 의도된 수신자에게 올바르게 배달되었음을 확인하는 배달 증명 시스템을 구현하였다. 그리고 이 시스템은 자바 암호 API를 기반으로 구현하였다.

An Implementation of E-Mail System with Certification of Delivery based on Java

Joon Woo[†] · Young-Guk Ha[††] · Shin-Young Lim[†††] · Jae-Kwang Lee[††††]

ABSTRACT

E-mail system is the most important service that enterprises and normal users in internet use. However, because a data security is not satisfied yet, an E-mail system with security service is essential. In this paper, We implemented the E-mail system with Certification of Delivery that was not provided in prior mail system with basic security services and can prove that sender's document is properly sent to the intended receipt. And an implementation of the system used Java Cryptography API.

1. 서 론

전자메일은 인터넷이 보편화됨에 따라 학술·연구용의 범위를 벗어나 기업과 일반 사용자들이 인터넷에서 사용하는 서비스 중 가장 중요한 서비스가 되었다. 전자메일은 시간과 공간을 초월하는 서비스로 사용범위가 점차 확대 되어가고 있지만, 전송되고 있는 데이터에 대한 보호 및 안전성은 기대에 미치지 못하고 있는 실정이다[1].

† 준 회 원 : 한남대학교 대학원 컴퓨터공학과
†† 정 회 원 : 한국전자통신연구원 연구원
††† 정 회 원 : 한국전자통신연구원 선임연구원
†††† 종신회원 : 한남대학교 컴퓨터공학과 교수

논문접수 : 1999년 10월 6일, 심사완료 : 1999년 11월 6일

메시지의 부당한 노출 및 변조는 일반 개인에게는 프라이버시의 침해가 될 수 있으며, 비밀성을 요구하는 특정한 사용자에게는 심각한 위협이 될 수 있다. 이러한 위협으로부터 사용자 정보를 안전하게 보호하기 위해서는 메시지 발신자의 신원에 대한 믿음을 제공할 수 있는 인증(Authentication) 기술과 안전하게 보호되어 전송되었음을 확인할 수 있는 기밀성(Confidentiality), 송·수신이 이루어진 후에 발생할 수 있는 부인행위에 대한 부인방지(Non-Repudiation) 등과 같은 정보보호 서비스를 제공할 수 있는 전자 메일 시스템이 요구된다[2].

이러한 정보보호 서비스를 제공하는 전자메일 시스템으로는 세계적으로 가장 많이 사용되는 보안 전자 메

일 시스템인 PGP(Pretty Good Privacy)가 있다. PGP는 RSA와 IDEA 키를 사용해 암호화하며 국내보다는 외국에서 널리 이용되고 있다. 그리고 전체 시스템의 구성이 복잡해서 현재는 널리 사용되고 있지 않는 PEM (Privacy Enhanced Mail)이 있으며, 최근에는 SSL, SET 등의 암호 알고리즘을 사용한 전자메일 시스템들이 등장하고 있다.

본 논문에서는 기본적인 정보보호 서비스 외에 기존의 메일 시스템에서는 제공되지 않는 발신자의 문서가 의도된 수신자에게 올바르게 배달되었음을 확인하는 배달 증명 시스템을 설계·구현하였다. 그리고 시스템의 구현은 자바 암호 API 클래스를 기반으로 하였다.

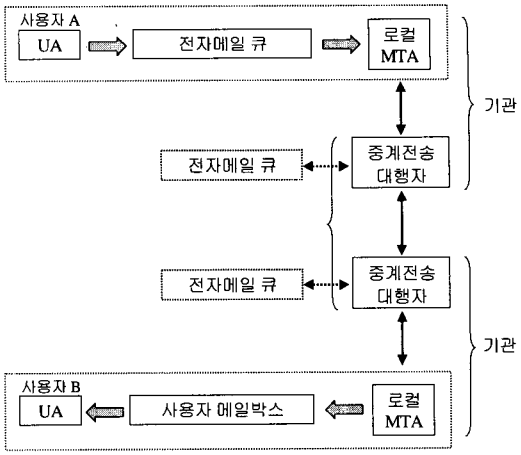
논문의 구성은 2장에서는 전자메일 시스템의 기본 구조와 전자메일의 정보보호 서비스를 기술하고, 3장에서는 배달증명 서비스를 분석하고 배달증명 기법을 제안하였다. 그리고 4장에서는 제안된 배달증명을 제공하는 전자메일 시스템을 설계하고, 5장에서는 배달증명이 가능한 전자메일 시스템을 구현하였다.

2. 전자 메일 정보보호 서비스

2.1 전자 메일 시스템의 구조

전자메일 시스템은 전자메일 메시지와 이것을 전달하는 MTA(Mail Transfer Agent)로 구성되고, 전자메일 메시지는 봉투(envelope), 헤더(header), 본문(body)으로 구성된다. 봉투는 모든 수신자에게 공통으로 적용되는 정보로 구성된 메시지 필드와 각 수신자마다 달리 적용되는 정보로 구성된 수신자별 필드로 나누어진다. 헤더는 본문 형태를 정의한 정보로 구성되고, 본문은 음성, 영상, 텍스트 등의 다양한 데이터를 포함한다. 전자메일 시스템은 (그림 1)과 같이 여러 개의 UA (User Agent)와 MTA(Mail Transfer Agent) 등으로 구성되며, 사용자 A가 사용자 B에게 메일을 전송하려 한다면 다음과 같은 과정을 거치게 된다[2].

① 사용자 A는 UA에서 메일 메시지를 작성하여 로컬 MTA에 메시지를 전달한다.
② 로컬 MTA에 전달된 메일 메시지는 먼저 메일 큐에 저장된 후, 중계전송 대행자들을 거쳐 최종 목적 MTA에 도달한다.
③ 사용자 B는 로컬 MTA의 수신된 메시지를 확인하고, 이 메시지를 사용자의 UA로 가져온다.



(그림 1) 인터넷 전자메일 시스템

- UA
  사용자와 메시지 전송 시스템 사이의 인터페이스를 제공하여 주며, 메시지의 전송을 위한 준비 및 전송된 메시지의 화면 출력을 가능하게 한다.
- MTA
  메시지 전송 시스템을 구성하는 프로세스로서 메시지의 전송 기능을 수행한다.

위와 같은 인터넷 전자메일 시스템은 메시지 전송에 있어서는 효율적이며 신뢰성은 있지만, 메시지의 불법 누출, 불법 변조, 메시지 송·수신자의 신원 조작, 송·수신 사실의 부인 등이 가능하며, 인터넷의 광범위한 특성상 송·수신자의 신원을 정확히 확인하기가 어렵다[1].

2.2 전자메일 정보보호 서비스

인터넷 전자 메일 환경에서는 앞 절에서 기술한 바와 같이 보안상의 취약점에 노출될 수 있다. 이러한 공격에 대처할 수 있는 정보보호 서비스는 다음과 같다[2, 7].

(1) 내용 기밀성(Content Confidentiality)

기밀성은 권한이 없는 사용자들에게 메시지가 노출되어지는 것을 막는 것을 의미한다. 즉, 발신자가 원하는 수신자만이 메시지를 확인할 수 있다. 두 시스템 사이에 가상회선이 개설되었다면 그 가상회선 상에 전송된 모든 사용자 자료를 공개되지 않도록 보호한다. 기밀성의 또 다른 특징은 트래픽 흐름분석에 대한 보

호로서 전송 자료의 출처와 목적지, 횟수, 길이 또는 통신선로상의 트래픽 특성에 대하여 공격자가 알지 못하도록 한다. 이와 같은 기밀성 서비스는 대칭 또는 비대칭 알고리즘을 사용하여 제공할 수 있다.

(2) 내용 무결성(Content Integrity)

기밀성 서비스와 같이 무결성 서비스는 메시지 스트림, 단일 메시지, 또는 메시지 특정 필드에 적용될 수 있다. 가장 유용한 접근방법은 메시지 스트림 전체를 보호하는 것이다. 메시지 스트림을 대상으로 하는 연결형 무결성 서비스는 메시지가 원래 송신된 대로 즉, 복사, 추가, 수정, 순서변경 또는 재 전송되지 않고 수신됐음을 확인한다. 자료에 대한 손상여부도 무결성 서비스 의해 제공된다.

(3) 메시지 발신자 인증(Message Origin Authentication)

발신자 인증은 통신이 신뢰성을 갖도록 보증한다. 발신자 인증 서비스는 메시지가 자기라고 주장하는 실체의 출처로부터 전송되었음을 수신자에게 확인시키는 서비스이다. 이 같은 서비스는 전자서명으로 제공할 수 있다. 전자서명은 발신자의 개인키를 획득하지 않는 한 위조될 수 없다. 전자서명을 가진 메시지는 일반적으로 재 사용되지 않는다.

(4) 발신 부인 방지(Non-repudiation of Origin)

메시지가 수신됐을 때, 수신자가 그 메시지가 실제로 송신자에 의해서 송신됐음을 확인할 수 있게 한다.

(5) 수신 부인 방지(Non-repudiation of Receipt)

메시지를 송신한 후에, 송신자가 실제로 수신자에 의해서 이 메시지가 수신됐었다는 것을 확인할 수 있게 한다. 이러한 수신 부인방지 서비스의 예로 배달증명과 내용 증명 서비스가 있다.

- 배달 증명(Certification of Delivery)
  부인방지 서비스 중의 하나로써 컴퓨터 통신망을 통해서 주고받는 전자문서에 대해 문서가 올바르게 의도된 수신자에게 배달되었음을 증명해주는 서비스이다.
- 내용 증명(Certification of Content)
  발신자가 수신자에게 어떤 내용의 메시지를 언제 발송하였다는 사실을 증명해주는 서비스이다.

3. 배달 증명 서비스

배달 증명 서비스는 내용증명 서비스와 연계되어 제공되는 보안 서비스로 현행 우편제도하에서의 특수 우편물 취급 서비스가 그대로 적용될 수 있다. 이들 서비스를 전자메일에서 제공하기 위한 방식은 아래와 같이 크게 직접 방식과 조정자 이용방식으로 구분된다[3]. 그리고 본 논문에서는 이 두 가지 방식의 장·단점을 비교 분석하여 간편하게 배달 증명 서비스를 제공할 수 있는 방식을 제안하였다.

(1) 직접방식

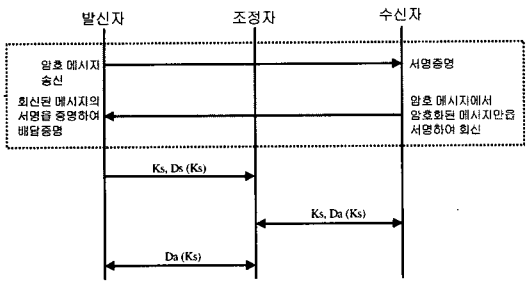
UA가 MTA를 경유하여 직접 다른 UA에게 메일을 전달하는 방식으로, 대표적인 모델은 MHS이다. 이러한 배달증명 서비스를 제공하기 위하여 많은 시도가 있어왔지만, 수신 증명 서비스를 제공하려고 할 경우 공평한 관계가 이루어지지 못하며 어느 한쪽이 이익을 보게된다.

(2) 조정자 이용방식

수신자와 송신자 사이에 모든 참가자가 신뢰할 수 있는 제 3의 조정자를 선정하여 모든 전자 메일이 이를 통하여 행하여지는 것으로 직접 방식의 문제점을 근본적으로 해결할 수 있다. 그러나 조정자가 필요하므로 이로 인한 프로토콜의 증가로 오버헤드가 증가하고 시스템의 구성이 복잡해진다.

(3) 제안된 방식

조정자를 이용한 방식은 프로토콜의 증가로 사용자에게 부담을 가중시켜 실질적인 사용을 저해하는 요인이 될 수 있다. 이러한 단점을 해결하기 위해서 프로토콜을 복잡하게 하는 내용증명 서비스를 제안된 방식에서는 제거하고, 배달증명 서비스만을 직접 방식을 이용해서 제공하기로 한다. 제안된 방식은 조정자 이용방식 중 공평한 부인방지 프로토콜로 가장 최근에 소개된 ZG(Zhou and Gollmann) 방식[3]을 응용하여 내용증명 서비스를 제공하는 프로토콜 부분을 제거하고 배달 증명 서비스를 제공하는 프로토콜만을 수용하였다. (그림 2)는 조정자를 이용한 ZG 방식을 응용하여 제안한 배달 증명 프로토콜을 기술하고 있다.

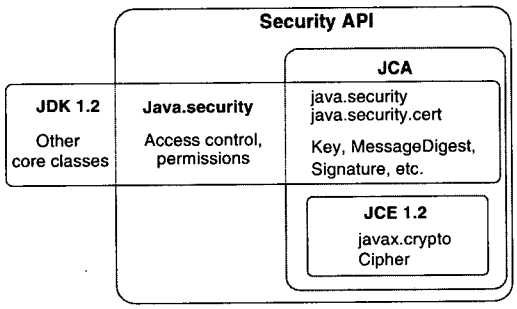


(그림 2) 제안된 배달증명 프로토콜

4. 전자 메일 시스템의 설계

4.1 자바 암호 패키지

자바 플랫폼에서는 자바 암호 패키지를 이용하여 쉽게 보안 서비스를 구현할 수 있다. 자바 암호 패키지는 Java Security API 소프트웨어 구조를 기반으로 하며, (그림 3)과 같은 기본 구성요소들을 포함하고 있다. 암호 클래스의 전반적인 설계는 JCA(Java Cryptography Architecture)를 기반으로 한다. JCA는 설계 패턴, 암호 컨셉들, 알고리즘을 정의하기 위한 설계 패턴, 그리고 확장된 구조를 명시한다. JCA는 구현 시 암호 컨셉을 분류하기 위해서 설계되었다. JCE(Java Cryptography Extension)는 JCA의 확장이며, SunJCE라는 다른 암호 Provider를 포함하고 있다. JCE는 핵심 JDK의 일부분이 아니라 JDK와 함께 동작하는 패키지로 추가하여 사용할 수 있는 표준 확장 라이브러리이다[7, 9].



(그림 3) JCA 모듈

4.2 암호화 모델

본 전자 메일 시스템에서 보안 서비스를 제공하기 위한 암호 모델은 아래에 제시된 알고리즘들을 기반으로 한다. 특히 ElGamal 알고리즘은 자바 암호 패키지에서 기본적으로 제공되지 않는 알고리즘으로 ElGamal 알고리즘 클래스들을 새롭게 작성하고, JDK 디렉토리의 lib/security 디렉토리에 있는 java.security 파일에 다음과 같이 ElGamal 알고리즘을 위한 새로운 provider를 삽입하였다.

security.provider.3=hannam.crypto.Provider

그리고 새로이 등록된 Provider 클래스에는 다음과 같이 ElGamal 알고리즘과 이의 구현 클래스에 대한 매핑들을 put 명령어를 사용하여 추가하였다.

```
put("KeyPairGenerator.ElGamal", "hannam.crypto.ElGamal-
    KeyPairGenerator");
put("Cipher.ElGamal", "hannam.crypto.ElGamalCipher");
put("Signature.ElGamal", "hannam.crypto.ElGamalSignature");
```

(1) 메시지 암호화 알고리즘 : DES 알고리즘

DES 알고리즘은 널리 사용되는 관용 암호 알고리즘이며, 데이터를 56비트 키를 이용하여 64비트 출력으로 변환한다. 관용 암호 알고리즘이기 때문에 암호화와 복호화에 사용되는 키가 하나이고 속도가 빠르다.

(2) 전자 서명 알고리즘 : ElGamal, MD5 알고리즘

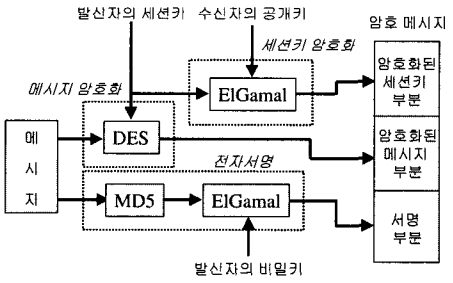
ElGamal 알고리즘은 유한체에서의 이산대수 계산의 어려움에 기반을 두고 있으며, 전자 서명과 암호화에 사용되는 공개키 암호 알고리즘이다. MD5 알고리즘은 MIT의 Ron Rivest가 개발한 것으로 임의의 길이의 메시지를 입력으로 받아들여서 일정한 길이의 비트열(메시지 다이제스트)을 출력하는 해쉬함수이다.

(3) 세션키 암호화 알고리즘 : ElGamal 알고리즘

4.2.1 메시지 암호화

발신자가 메시지를 암호화하여 생성하는 암호 메시지는 암호화된 세션키 부분, 암호화된 메시지 부분, 그리고 서명 부분으로 구성된다. (그림 4)는 다음에 기술한 메시지 암호화 과정을 거쳐 암호 메시지를 생성하는 과정 및 구현 코드를 보여주고 있다.

① 메시지를 DES 알고리즘을 사용해서 발신자의 세션키로 메시지를 암호화한다. 그리고 진하게 강조된 글자는 사용된 알고리즘을 보여주고 있다.



(그림 4) 메시지 암호화

```
byte[] bodyPlaintext = body.getBytes();
Cipher cipher = Cipher.getInstance("DES");
cipher.init(Cipher.ENCRYPT_MODE, sessionKey);
byte[] bodyCiphertext = cipher.doFinal(bodyPlaintext);
```

② 메시지 암호화에 사용된 DES 세션키를 ElGamal 알고리즘을 사용해서 암호화한다.

```
cipher = Cipher.getInstance("ElGamal");
cipher.init(Cipher.ENCRYPT_MODE, theirPublicKey);
byte[] sessionKeyCiphertext = cipher.doFinal(sessionKey.-
  getEncoded());
```

③ MD5 알고리즘을 사용해서 메시지 다이제스트를 생성하고, 이 메시지 다이제스트를 ElGamal 알고리즘으로 암호화하여 전자 서명을 생성한다.

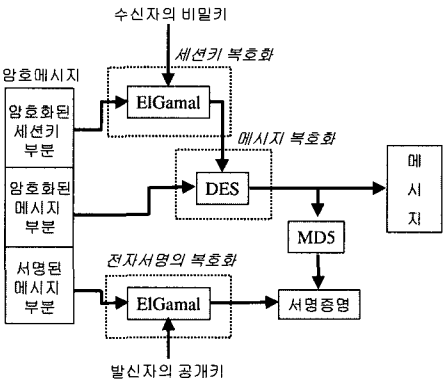
```
MessageDigest md = Mes sageDigest.getInstance("MD5");
md.update(bodyPlaintext);
byte[] msg_digest = md.digest();
Signature s = Signature.getInstance("ElGamal");
s.initSign(ourPrivateKey);
s.update(msg_digest);
byte[] bodySignature = s.sign();
```

4.2.2 메시지 복호화

수신된 암호 메시지는 메시지의 각 부분별로 복호화 및 서명을 증명하는 데, (그림 5)는 이러한 과정을 보여주고 있다.

① ElGamal 알고리즘을 사용해서 DES 세션키를 복호화 한다.

```
PrivateKey ourPrivateKey =
  mKeyManager.getPrivateKey();
Cipher cipher = Cipher.getInstance("ElGamal");
cipher.init(Cipher.DECRYPT_MODE, ourPrivateKey);
byte[] sessionKeyPlaintext = cipher.doFinal(sessionKey-
  Ciphertext) ;
```



(그림 5) 메시지 복호화

```
SecretKeyFactory skf = SecretKeyFactory.getInstance-
  ("DES");
DESKeySpec desSpec = new DESKeySpec(sessionKey-
  Plaintext, 0);
SecretKey sessionKey = skf.generateSecret(desSpec);
```

② DES 알고리즘을 사용해서 복원된 DES 세션키를 가지고 암호문을 복호화 한다.

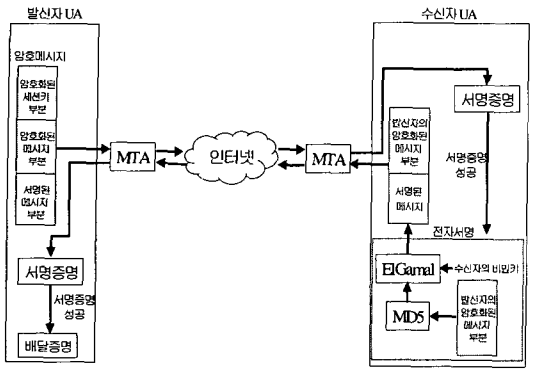
```
cipher = Cipher.getInstance("DES");
cipher.init(Cipher.DECRYPT_MODE, sessionKey);
byte[] plaintext = cipher.doFinal(bodyCiphertext);
```

③ 복원된 평문 메시지를 MD5 알고리즘을 사용해서 메시지 다이제스트로 변환하고, ElGamal 알고리즘을 사용해서 전자 서명을 증명하게 된다.

```
MessageDigest md = MessageDigest.getInstance("MD5");
md.update(plaintext);
byte[] msg_digest = md.digest();
Signature s = Signature.getInstance("ElGamal");
s.initVerify(theirPublicKey);
s.update(msg_digest);
if (s.verify(bodySignature)) {
    new MessageBox(this, "서명 증명 성공", "서명이 올
      바르게 증명되었습니다");
}
else {
    new MessageBox(this, "서명 증명 실패", "서명이 올
      바르지 않습니다");
}
```

4.3 배달증명 모델

(그림 6)은 앞에서 제안된 배달증명 방식을 기반으로 한 배달증명 모델을 통해서 의도된 수신자가 올바르게 메일 메시지를 수신하였음을 확인하는 과정을 보



(그림 6) 배달증명 모델

여주고 있다. 그리고 암호화된 메시지의 서명을 위해서는 암호 모델에서와 같이 MD5와 ElGamal 알고리즘을 사용한다. 특히 본 논문에서 서명 알고리즘으로 RSA 알고리즘을 사용하지 않고 ElGamal 알고리즘을 사용한 이유는 ElGamal 알고리즘은 1997년에 특허권이 종료되어 현재 자유롭게 사용할 수 있지만, RSA 알고리즘은 2000년까지 특허권이 보장되고 있다는 것이다. 그리고 향후 배달 증명을 제공하기 위한 방식으로 현재 국내에서 연구가 진행되고 있는 ElGamal 알고리즘을 통한 불확정 전송 프로토콜을 적용 하고자 한다. 불확정 전송 프로토콜은 송신자가 수신자에게 어떤 비밀정보를 보내고자 할 때 수신자는 그 비밀정보를 1/2의 확률로 취할 수 있게 하고, 송신자는 수신자가 그 비밀정보를 취했는지 여부를 1/2의 확률로 추측할 수 있게 하는 프로토콜로써 전자 우편에 적용되어 공평한 배달증명 모델을 설계하는 데 도움을 줄 것으로 기대된다.

① 발신자가 배달증명 서비스를 요청했을 때, 메시지 암호화와 동일한 과정으로 암호 메시지를 생성하고, 이 메시지에 다음과 같이 배달증명을 요청하는 플래그를 함께 붙여서 수신자에게 송신한다.

```
String unbroken = "Certification of delivery:" + base64.-
  encode(plaintext);
```

② 수신자 쪽의 UA는 수신된 메시지에서 플래그를 확인하여 배달증명 요청 플래그가 있다면, 전자 서명을 증명하고 암호화된 메시지를 수신자의 비밀키를 사용하여 서명한 후에 발신자의 암호화된 메시지와 함께 발신자에게 회신한다.

```
if (return_flag == 1) {
   PrivateKey ourPrivateKey = mKeyManager.getPrivateKey();
   MessageDigest md = MessageDigest.getInstance("MD5");
   md.update(bodyCiphertext);
   byte[] msg_digest = md.digest();
   Signature s = Signature.getInstance("ElGamal");
   s.initSign(ourPrivateKey);
   s.update(msg_digest);
   byte[] bodySignature = s.sign();
   ......
}
```

③ 발신자는 회신된 메시지의 전자서명을 증명하여 의도된 수신자에게 메일이 배달되었음을 확인한다.

```
if (receipt_flag == 1) {
   ......
   Signature s = Signature.getInstance("ElGamal");
   s.initVerify(theirPublicKey);
   s.update(msg_digest);
   if (s.verify(bodySignature)) {
      new MessageBox(this,"***** 서명이 올바르게 증명
     되었습니다 ! *****\n", "*****[배달" + "증명]\n" +
     theirName + "가 올바르게 메일을 수신하였음을 확인합
     니다.****\n");
   }
   else {
      new MessageBox(this,"서명 증명 실패!","\n" +
     "*****배달증명에 실패하였습니다." + "*****");
   }
   ......
}
```

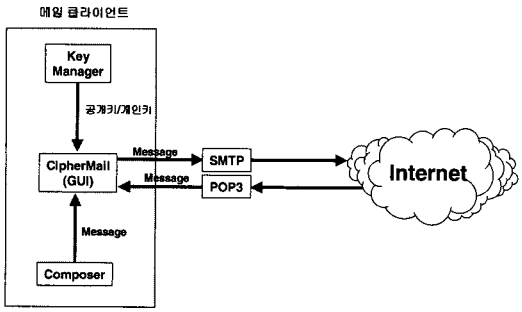
5. 전자 메일 시스템의 구현

5.1 모듈 구성

본 전자메일 시스템은 메일 클라이언트로 제공되며, 메시지의 암호화와 복호화 및 배달증명을 담당하는 CipherMail 클래스와, 메일 메시지를 저장하고 관리하는 Message 클래스를 핵심으로 하고 있다. 이와 같이 메일 클라이언트를 기반으로 하기 때문에 MTA의 변경 없이 정보보호 서비스가 가능하며, 메일 메시지를 암호화 및 서명하여 전송함으로써 메시지가 네트워크상의 전송 과정에서 도청되거나, 불법 변조되는 등의 보안상의 문제에 효율적으로 대처할 수 있게 되었다. (그림 7)은 Message 클래스를 사용하는 CipherMail 애플리케이션 모듈을 구성하는 클래스들의 구성도이다.

(1) POP3(메일 수신 클래스)

POP3 메일 서버로부터 수신된 메시지를 확인하고 사용자의 메일 프로그램으로 메일 메시지를 가져오는



(그림 7) 전자메일 시스템 모듈

역할을 한다.

(2) SMTP(메일 송신 클래스)

설정된 SMTP 메일 서버에 메일 메시지를 전달하는 역할을 한다.

(3) Composer(메일 작성 클래스)

새로운 메시지를 작성하는 윈도우를 생성하고, 수신자의 메일 주소와 수신자의 공개키를 선택하고 메일 제목과 본문을 입력하여 메일 메시지를 작성할 수 있게 한다.

(4) CipherMail(메인 메일 윈도우 클래스)

메일 메시지 리스트를 관리하고 메시지의 내용을 출력하는 메인 애플리케이션 윈도우이다. 그리고 Composer 클래스에 의해서 생성된 메시지를 암호·복호화 및 배달 증명하는 기능을 한다.

(5) KeyManager(키 생성 및 공개키 관리 클래스)

ElGamal 키 쌍을 생성하고 메시지를 송신하고자 하는 상대방 수신자의 공개키를 관리하는 역할을 한다. 다음은 이러한 역할을 수행하는 create 메소드, getPublicKey 메소드, getPrivateKey 메소드를 보여주고 있다.

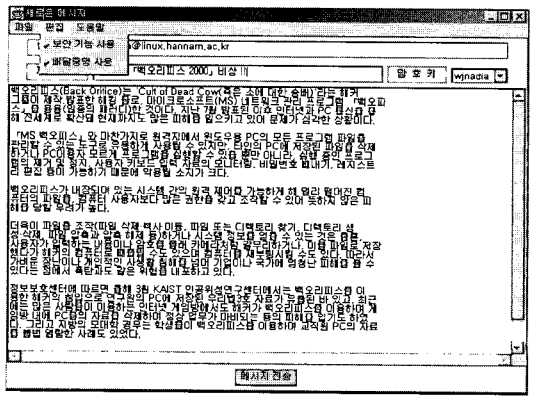
```
public static KeyManager create(String file, String name,
  KeyPair pair) {
    KeyManager km = new KeyManager(name, pair);
    km.mKeyFile = file;
    return km;
}

public synchronized PublicKey getPublicKey(String
  name) {
    if (name.equals(getName())) return getPublicKey();
    return getIdentity(name).getPublicKey();
}

public PrivateKey getPrivateKey() { return mPrivateKey;
  }
```

5.2 구현 결과

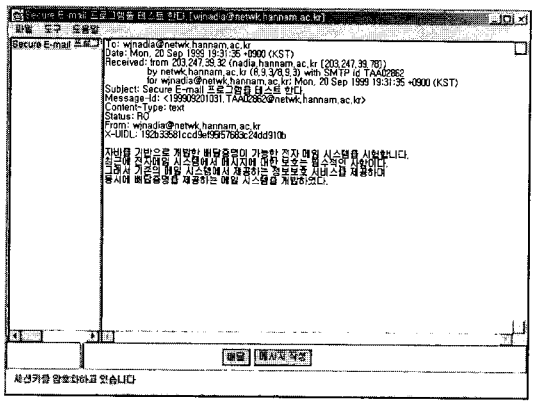
5.2.1 메시지 작성 및 송신

메시지를 작성하기 위해서는 시작 윈도우의 메뉴에서 메시지 작성 메뉴 혹은 버튼을 선택한다. 메일 메시지는 크게 세 가지 모드로 전송될 수 있다. 첫 번째는 보안 기능과 배달증명을 사용하지 않고 평문 그대로를 전송하는 모드이다. 어떤 보안 기능도 사용하지 않으므로 보안상의 취약점을 지니게 되지만, 빠르게 메일을 전송할 수 있다. 두 번째는 보안 기능을 사용하는 것으로 메일 메시지에 대한 암호화 및 서명 등을 제공하여 안전하게 수신자에게 메일을 전송할 수 있다. 세 번째는 보안 기능과 배달 증명 서비스를 사용하는 것으로 기본 정보보호 서비스를 제공할 뿐만 아니라, 의도된 수신자가 메일을 올바르게 수신하였음을 확인할 수 있도록, 수신자가 서명한 메시지를 회신하게 된다.

(그림 8)에서는 메시지 작성 윈도우의 메뉴 바에서 '보안 기능 사용' 메뉴와 '배달 증명 사용' 메뉴를 선택하여 기본 보안 기능과 배달증명 서비스를 제공받을 수 있도록 설정하였다. 마지막으로 메시지 전송 버튼을 누르면 (그림 9)와 같이 메일 메시지의 암호화 및



(그림 8) 메시지 작성



(그림 9) 메시지 암호화 및 서명

서명 등의 과정을 거쳐 의도된 수신자에게 메시지를 전송한다.

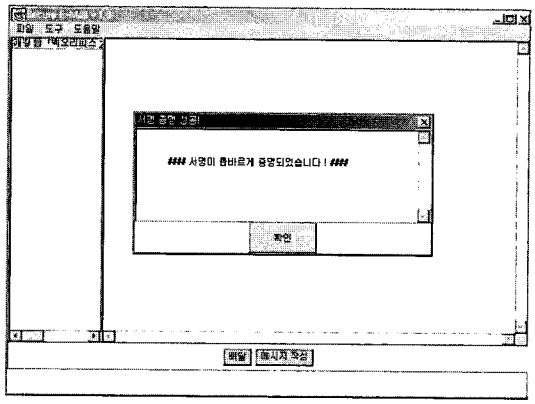
(그림 10)은 메시지 암호화 및 서명을 통해서 생성된 암호 메시지 자체를 출력한 것으로 배달 증명을 요청하기 위해서 "Certification of delivery:" 태그를 암호 메시지의 맨 처음에 첨부하였다.

Certification of delivery:AAd3am5hZGlhAAAAg
xuZrsq5+dATONIEZxAiqudx7ppYs+74RARd9H8L6
rmjPTuyaDMpm+SOE+pLUI4+BNK2FfwawgzcjdgWG
GrrdBvN7Iz18yDhazbTIYsdvA0MtRxsnB5y7dojk
uKVQf8R093q27yVGW4VAAP/3plPgYXxJuYLAAAAg
Fur8S7UYEW7aU5qh556yjwvHSmZUiZ3Yxyg2ZNRf
9VKR6Iz22ZBDJAVmEHNNr9L+CYFRgKIm/yxfycwC
.....

(그림 10) 암호 메시지

5.2.2 메시지 수신 및 서명 증명

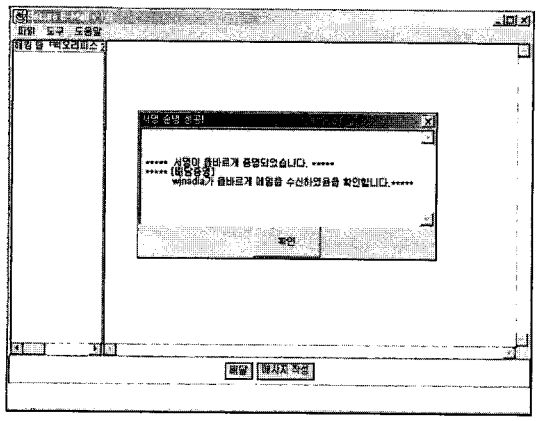
메시지를 수신한 수신자의 메일 프로그램은 배달 증명을 요구하는 메시지임을 메시지 내의 플래그를 인지하여 확인하고, 서명을 증명하여 (그림 11)과 같이 서명이 올바르게 증명되었음을 다이얼로그 박스에 출력한다. 그리고 수신된 메시지 중에서 암호화된 메시지만을 서명하고 발신자의 암호화된 메시지와 함께 "Receipt:" 플래그를 메시지의 처음에 첨부하여 발신자에게 회신한다.



(그림 11) 수신된 메시지의 서명 증명

5.2.3 메시지 회신 및 배달 증명

회신 메시지를 수신한 발신자는 메시지에 붙어있는 플래그가 배달 증명에 대한 회신 메시지를 나타냄을 확인한다. 그리고 수신된 메시지의 서명을 증명하여 배달증명이 확인되었음을 나타내는 다이얼로그 박스를 (그림 12)와 같이 출력한다.



(그림 12) 회신된 메시지의 배달증명

5.3 비교 분석

본 논문에서 구현한 메일 시스템은 메시지 기밀성, 메시지 무결성, 송신자 인증, 송신자 부인 봉쇄 서비스 등을 제공하며, PGP와 PEM에서 아직까지 제공하지 않고 있는 배달증명 서비스를 제공한다. <표 1>은 본 논문에서 구현한 메일 시스템과 PGP 및 PEM에서 제공하고 있는 정보보호 서비스에 대한 비교를 기술하고 있다.

〈표 1〉 PGP 및 PEM과 제안된 메일 시스템 비교

정보 보호 서비스	제안된 시스템	PGP	PEM
메시지 기밀성	제공함	제공함	제공함
메시지 무결성	제공함	제공함	제공함
송신자 인증	제공함	제공함	제공함
송신부인봉쇄	제공함	제공함	제공함
배달증명	제공함	제공하고 있지 않음	제공하고 있지 않음

6. 결 론

현재의 범세계적인 네트워크 환경에서 광범위하게 사용되고 있는 전자메일 시스템은 많은 보안상의 취약점에 노출되어 있다. 이러한 전자메일의 취약점을 극복하기 위해서 다양한 보안 메일 시스템들이 소개되고 있지만 사용자에게 만족스런 서비스를 제공하지 못하고 있다. 본 논문에서는 기존의 메일 시스템에서 제공되는 기본 보안 서비스를 제공하며, 의도된 수신자가 메시지를 올바르게 수신하였음을 증명하는 배달증명 서비스를 제공하는 보안 메일 시스템을 설계·구현하였다. 구현은 자바 암호 API를 기반으로 하였으며, 이를 포함한 자바 플랫폼은 네트워크 및 보안 서비스를 제공하는데 필수적인 모든 요소들을 클래스로 갖추고 있기 때문에 개발자에게 프로그램의 작성을 용이하게 한다.

향후 보안 메일 시스템에서는 배달증명 뿐만 아니라, 내용증명까지도 상호간에 공평하고 간편하게 제공할 수 있는 방법을 모색해야 할 것이다.

참 고 문 헌

[1] 강명희, "인터넷 메일 시스템에서의 정보 보호 서비스 구현", 광운대학교 전자계산학과 석사학위 논문, 1995.

[2] 조한진, 김봉한, 이재광, "정보보호 서비스를 위한 Secure E-mail 시스템 설계", 한남대학교 산업기술연구소, 1998.

[3] 박춘식, "배달 및 내용 증명이 가능한 전자메일", 통신정보보호학회지, 제7권 제2호, 1997. 6.

[4] 손진욱 편저, "Java 2 Programing Bible", 정보문화사, 1999.

[5] 이재용, 이기수, 장춘서, "PGP를 이용한 WWW 메일 시스템의 설계 및 구현", 한국정보과학회 가을 학술발표논문집, 제24권 제 2호, 1997.

[6] 홍주영, 윤이중, 김대호, "전자우편 시스템의 보호 방식 분석", 통신정보보호학회지 Vol.4 No.2, 1994. 6.

[7] 최용락, 소우영, 이재광, 이임영, "통신망 정보 보호", 그린출판사, 1995.

[8] Jonathan Knudesen, "Java Cryptography," O' REILLY, 1998.

[9] Sun Microsystems, "Java 2 SDK, Standard Edition Documentation," 1999.

[10] Scott Oaks, "Java Security," O' REILLY, 1998

[11] Elliotte Rusty Harold, "Java Network Programing," O' REILLY, 1997.

[12] Bruce Schneier, "Applied Cryptography," John Wiley & Sons Inc., 1996.

[13] J. Zhou and D. Gollmann, "A Fair Non-repudiation Protocol," Proc. of the 1996 IEEE Symposium on Security and Privacy, 1996.

[14] J. Zhou and D. Gollmann, "Observations on Non-repudiation," Advances in Cryptology, Proceedings of ASIACRYPT '96, Spring-Verlag, 1996.

우 준

e-mail : wjnadia@netwk.hannam.ac.kr

1998년 한남대학교 컴퓨터공학과 졸업(학사)

1999년~현재 한남대학교 대학원 컴퓨터공학과 석사과정

관심분야 : 컴퓨터통신보안, 컴퓨터네트워크

하 영 국

e-mail : ygha@econos.etri.re.kr

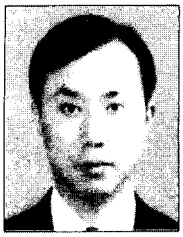
1993년 건국대학교 전산학과 졸업(학사)

1995년 건국대학교 대학원 전산학과 졸업(석사)

1995년~1998년 한국전자통신연구소 부설 시스템공학연구소 연구원

1998년~현재 한국전자통신연구원 전자상거래연구부 연구원

관심분야 : 전산망 보안, PKI, 전자상거래, 컴퓨터 통신



임 신 영

e-mail : sylim@econos.etri.re.kr

1992년 건국대학교 전산학과 졸업 (석사)

1997년 고려대학교 컴퓨터과학과 (박사 수료)

1996년~현재 한국전자통신연구원 전자상거래연구부 선임연구원

관심분야 : 전산망 보안, PKI, 키복구, 생체인식

이 재 광

e-mail : jklee@netwk.hannam.ac.kr

1984년 광운대학교 전자계산학과 학사

1986년 광운대학교 대학원 전자계산학과 석사

1993년 광운대학교 대학원 전자계산학과 박사

1986년~1993년 군산전문대학 전자계산학과 부교수

1993년~현재 한남대학교 컴퓨터공학과 부교수

관심분야 : 컴퓨터네트워크, 정보통신정보보호