IE Interfaces
Vol. 13, No. 3, pp. 548-555, September 2000.

Design and Implementation of a Secure E-Mail System for Electronic Commerce Information Exchange

Shin-Young Lim[1] · Ho-Sang Ham[1] · Ok-Hwan Byeon[2] · Tai-Yun Kim[3]
[1]Electronics and Telecommunications Research Institute / [2]KORDIC
/[3]Department of Computer Science, Korea University

전자상거래 정보 교환을 위한 안전한 전자우편 시스템의 설계 및 구현

임신영 · 함호상 · 변옥환 · 김태윤

TAn E-Mail system is one of the most important services for enterprise and electronic commerce end users on the Internet. However, security for an E-Mail service is not satisfied yet, an E-Mail system with security service is definitely required especially in electronic commerce system. In this paper, an E-Mail system with confirmation of e-mail delivery is proposed. The certification of delivery of E-Mail message is not provided in conventional E-Mail systems. The proposed E-Mail system is composed of this certification of delivery and basic security services. The certification of delivery can prove sender's E-Mail message is securely sent to legitimate receivers. The system is designed and implemented by Java Cryptography API.

1. Introduction

As E-Mail service is popular, it covers not only academic and research purposes but also enterprise and electronic commerce services. An E-Mail system is one of very important service in electronic commerce because this service can be used as an infrastructure for other applications for electronic commerce services. But current E-Mail service does not provide end users with security services(Kang, 1995). Many kinds of security attacks in E-Mail service such as confidentiality, integrity, modification, and repudiation are the causes of privacy breaches and sensitive enterprise information disclosure. For protecting from these kinds of attacks, authentication for identification of an E-Mail originator, confidentiality for E-Mail message, and non-repudiation for delivery and contents of E-Mail are required for E-Mail system(Cho, Kim, Lee,1998). Currently, one of the popular secure E-Mail systems is PGP(Pretty Good Privacy)-based E-Mail system that uses cryptographic keys of RSA and IDEA. And PEM(Privacy Enhanced Mail) is also one of them but complex in architecture. And recently, SSL-based secure E-Mail systems are introduced in public. In this paper, an E-Mail system that an originator can prove delivery of an E-Mail message to a legitimate receiver as well as E-Mail message encryption, digital signature, and public key based encryption key exchange is designed and implemented. The system is implemented using Java Crypto-API and Sun JCE(Java Cryptography Extension) compatible cryptographic classes. The contents of this paper is composed of basic structure of an E-Mail system and security requirements of an E-Mail in chapter 2, the analysis and proposal of certification of delivery of an E-Mail message in chapter 3, the design of the system in chapter 4, and the implementation of the system in chapter 5.

2. Structure and Security Requirements of an E-Mail System

2.1 Structure of an E-Mail System

Briefly speaking, an E-Mail system is consisted of

an E-Mail message and a MTA(Mail Transfer Agent) that transmits an E-Mail message. And the E-Mail message is composed of an envelope, a header, and a body. And the envelope is composed of message field that can be commonly applied to each recipient and recipient field that can be applied to individual recipient. The header is filled with the definition of type of the body. And body is container of various type of data such as text, voice, image, and multimedia. As shown in <Figure 1>, a typical E-Mail system is composed of several UA(User Agent) and the process of transmitting a message from user A to user B as follows(Cho, Kim, Lee, 1998);

Step1 : User A writes a mail message and transmit it to a local MTA

Step2 : Then the mail message is stored in the message queue of MTA and MTA transmits the mail message to the destination MTA through mid-MTAs.

Step3 : User B, then, checks the incoming message of MTA, gets the mail message that was sent by user A.

- UA(User Agent)
  UA provides E-Mail end users with interface between user and message transfer agent such as graphic user interface for message transmission and transmitted message.

- MTA(Message Transfer Agent)
  MTA performs transmitting mail messages from subscribers to legitimate receiver. MTA also manages incoming and outgoing message queues.

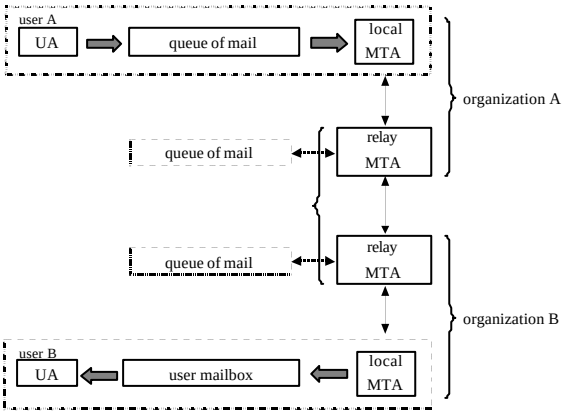

Figure 1. Typical Internet E-Mail System.

Internet E-Mail system provides end users with convenience but not provides security such as illegitimate message monitoring, modification, masquerading, and repudiation of mail messages and originators(Kang, 1995).

2.2 Security Requirements of the E-Mail System

For guaranteeing E-Mail service is secure, the following security service is required (Choi, So, Lee, 1995; Cho, Kim, Lee, 1998).

(1) Content Confidentiality

In the E-Mail context, a transmitted message is required to be read by a legitimated receiver. There is no points of eavesdrop or monitoring the contents or history of the transmitted message from the sender to the legitimated receiver. It is possible by using asymmetry and symmetry cryptographic algorithms.

(2) Content Integrity

Like content confidentiality, content integrity means contents of the transmitted mail message did not change while preparing message transmission, message transmitting, and receiving the transmitted message

(3) Authentication of Message Origin

The authentication of message origin provides legitimate receivers with assurance of identification of the sender, origin. It is possible by using digital signature.

(4) Non-repudiation of Origin

When the legitimate receiver receives the transmitted mail message, the receiver is required to verify the identification of the message origin.

(5) Non-repudiation of Receipt

After transmission of the mail message, the origin is required to verify the reception of the transmitted message by the legitimated receiver. The verifications of reception are certification of delivery and contents.

- Certification of Delivery
  It means the assurance of reception to the origin that the transmitted message is received only to the legitimate receiver. The point is the legitimate receiver himself gets the message.
- Certification of Contents
  It means the assurance of the time of the message reception of the legitimate receiver and contents

of the message to the receiver. The certification of content is for the sender.

3. Certification of Delivery

The service of certification of delivery is related with the certification of contents and it is similar to the registered special mail delivery service in post office. For applying this service to the E-Mail system, direct method and mediator method are required(Park, 1997. 6). In this paper, hybrid method of mixture of the direct method and the mediator method is proposed. The hybrid method is simple in certification of E-Mail delivery.

(1) Direct Method

The E-Mail message is transmitted by UA through his MTA to the MTA of the legitimate receiver directly. The MHS(Message Handling System) is considered as one of the direct method. Although there were many trials for certification of an E-Mail message delivery by the direct method, the proof of reception was not properly implemented because of unfair relationships between sender and receiver when proofing the E-Mail message delivery.

(2) Mediator Method

The mediator method for certification of an E-Mail message delivery is involving a trusted third party, a mediator, performing verification of E-Mail message transmission. But this method causes overhead of communication protocol and makes system more complicated.

(3) Hybrid Method

One of the major contributions of this paper is

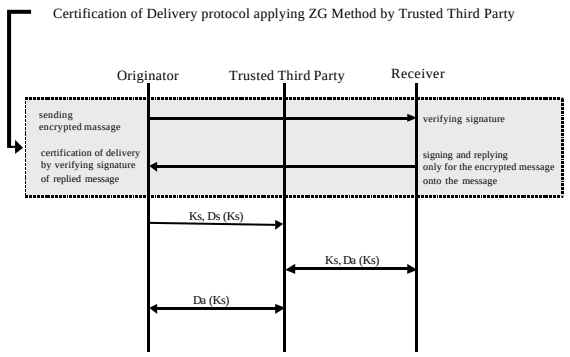

Figure 2. Proposed Certification of Delivery Protocol.

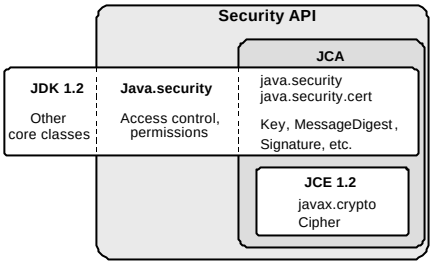

Figure 3. JCA Architecture.

proposition of hybrid method that is stems from modifying the weakness of the mediator method andreferencing the ZG(Zhou and Gollmann) method that is one of fair non-repudiation protocols(Park, 1997. 6). The hybrid method eliminates the certification of contents in the mediator method and emphasis on the certification of delivery.

4. Design of the E-Mail System

4.1 Java Environments

As Java is platform independent, it is useful for installing if the E-Mail System that provides the certification of E-Mail message delivery is implemented in Java environments. In this paper, the components of the primitive functions of the system are from Java cryptographic classes that are developed by the authors. As shown in <Figure 3>, the Java cryptography package is based on the Java Security API. The preparation of Java cryptographic classes is referred to the JCA(Java Cryptography Architecture). The JCE(Java Cryptography Extension) is the extension of the JCA and contains other cryptographic providers such as SunJCE. The JCE is the extension of library of de facto standard of cryptography(Choi, So, Lee, 1995; Sun Microsystems, 1999).

4.2 Cryptographic Modules

For providing the certification of delivery in E-Mail system, the Java cryptographic modules are designed. Especially, SEED and RSA cryptographic algorithms, not provided in Java cryptographic package, are developed and are inserted as a new provider in the file named java.security in the directory of jre/lib/security .

security.provider.3=developers.crypto.Provider

And the following information is adds using 'put' command for class mapping to these algorithms to the 'developers.java' file that is newly registered provider class.

```
put("KeyGenerator.SEED", "developers.crypto.
      provider. SEEDKeyGenerator");
put("SecretKeyFactory.SEED", "hannam.crypto.
      provider.SEEDSecretKeyFactory");
put("Cipher.SEED",
      "developers.  crypto.provider.SEED");
```

(1) Message Encryption Algorithm : SEED

Seed is a symmetric block cipher. The cipher has a block size of 128 bits and a key size of 128 bits. The cipher has been designed based on the Feistel structure.

The block cipher Seed has the following features:

- Global structure : Feistel cipher
- Block length : 128 bits
- Key length : 128 bits
- The number of rounds : 16 rounds

Seed can be used for providing confidentiality at electronic commerce in KOREA. SEED has been developed and has been proposed for standardization in KOREA.

(2) Digital Signature Algorithm : RSA with SHA-1

SHA-1 stands for Secure Hash Algorithm. It was developed by the NIST(National Institute of Standards and Technology) in conjunction with the NSA. Like MD5, SHA-1 is based on MD4. The changes made in SHA-1, however, are considerably different from the changes made in MD5. Also, SHA-1 produces a message digest value that is 160 bits long, which increases its resistance to attack.

(3) Session Key(Symmetric Key) Exchange Algorithm : RSA

RSA is one of the public-key cryptographic algorithms, and one of the most widely used ones. The public exponent and the modulus together are known as the public key; the secret exponent and the modulus together are known as the private key.

4.2.1 Message Encryption

The message encryption is consisted of a encrypted session key, a encrypted message, and a digital signature. <Figure 4> epicts the process of message encryption.

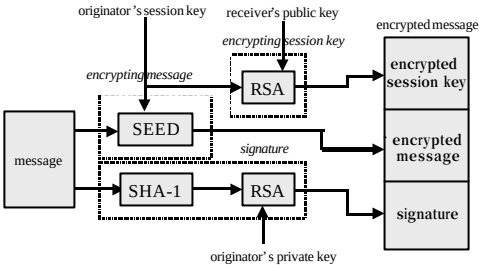

Figure 4. Diagram of Message Encryption.

① Using SEED encryption algorithm, E-Mail message is encrypted

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

② The SEED session key is encrypted by receiver's RSA public key exchange key and RSA algorithm.

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

③ Using SHA-1 algorithm, inputs the E-Mail message and gets a message digest. And using sender's RSA digital signature key and RSA algorithm, creates digital signature.

```
MessageDigest md = MessageDigest.getInstance("SHA-1");
md.update(bodyPlaintext);
byte[] msg_digest = md.digest();
Signature s = Signature.getInstance("RSA","DEVELOPERS");
s.initSign(ourPrivateKey);
s.update(msg_digest);
byte[] bodySignature = s.sign();
```

4.2.2 Message Decryption

The received message is decrypted three detailed parts as shown in <Figure 5>.

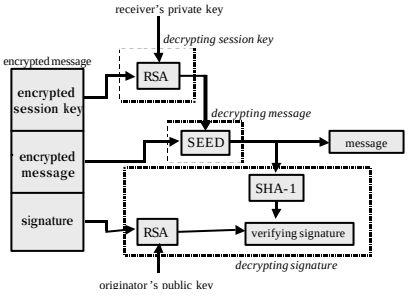

Figure 5. The Diagram of Message Decryption.

① Using RSA algorithm, decrypts SEED session key.

```
PrivateKey ourPrivateKey = mKeyManager.getPrivateKey();
Cipher cipher=Cipher.getInstance("RSA","DEVELOPERS");
cipher.init(Cipher.DECRYPT_MODE, ourPrivateKey);
byte[] sessionKeyPlaintext=cipher.doFinal(sessionKeyCiphertext);
SecretKeyFactory skf = SecretKeyFactory.getInstance
                                ("SEED","DEVELOPERS");
SEEDKeySpec seedSpec=new SEEDKeySpec
                                (sessionKeyPlaintext, 0);
SecretKey sessionKey = skf.generateSecret(seedSpec);
```

② Using SEED algorithm and decrypted SEED session key, decrypts encrypted message.

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

③ Using SHA-1 algorithm and decrypted message, gets a message digest. And verify digital signature value using RSA algorithm.

```
MessageDigest md=MessageDigest.getInstance("SHA-1");
 md.update(plaintext);
 byte[] msg_digest = md.digest();
Signature s=Signature.getInstance("RSA","DEVELOPERS");
 s.initVerify(theirPublicKey);
 s.update(msg_digest);
 if (s.verify(bodySignature)) {
   new MessageBox(this, "Signature Certificate Success.)", " " +
       "### Signature Verified.) ###");
}
else {
   new MessageBox(this, "Signature Certificate Failed.)", " " +
       "### Signature Verified Failed) ### ");
}
```

4.3 Certification of Delivery Module

As shown in <Figure 6>, the message origin is able to verify the delivery of the transmitted message to the legitimate receiver. For digital signature, SHA-1 and RSA is used.

① When the message origin requests the certification of delivery, creates encrypted message same as 4.2.1 in this paper. And adds a flag of request for certification of delivery to the encrypted message. Then sends it to the legitimate receiver.

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

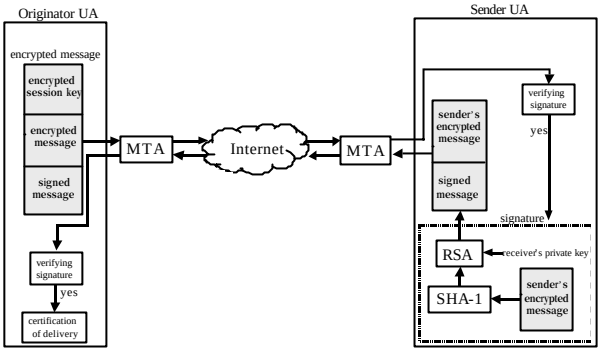

Figure 6. Certification of Delivery Module.

② When the message origin requests the certification of delivery, creates encrypted message same as 4.2.1 in this paper. And adds a flag of request for certification of delivery to the encrypted message. Then sends it to the legitimate receiver.

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

③ When UA of the legitimate receiver checks the flag of request for certification of delivery, verifies the digital signature, signs with his private digital signature key, replies to the message origin with the encrypted message.

```
if (return_flag == 1) {
 PrivateKey ourPrivateKey = mKeyManager.getPrivateKey();
 MessageDigest md = MessageDigest.getInstance("SHA-1");
  md.update(bodyCiphertext);
  byte[] msg_digest = md.digest();
 Signature s = Signature.getInstance("RSA","DEVELOPERS");
  s.initSign(ourPrivateKey);
  s.update(msg_digest);
  byte[] bodySignature = s.sign();

}
```

④ The message origin verifies the replied message with the digital signature verification for confirmation of certification of E-Mail message delivery.

```
if (receipt_flag == 1) {
 Signature s = Signature.getInstance("RSA","DEVELOPERS");
  s.initVerify(theirPublicKey);
  s.update(msg_digest);
  if (s.verify(bodySignature)) {
  new MessageBox(this,"Signature Certificate Success.)",
  " " +" ### [Certification of Delivery] \r\n" +
  " " +" Receiver read mail ### \r\n");
  }
```

```
else {
  new MessageBox(this,"Signature Certificate Failed!!",
   " " +" ### [Certification of Delivery Failed] ### \r\n"
 }
 ... ...
}
```

5. Implementation of the E-Mail system

5.1 Major Components

The implemented E-Mail system is provided in the form of E-Mail client. The major components are CipherMail class for message encryption/decryption and certification of delivery, and Massage class for storing and managing of mail message. For implemented in the form of E-Mail client, it is easy to install to conventional Internet E-Mail system and there is no need to change in the part of MTA. The <Figure 7>. describes the classes for CipherMail application module using Message class.

(1) POP3(Mail receiving class)
POP3 performs reception of incoming E-Mail messages and gets the received messages to the specific E-Mail client system

(2) SMTP(Mail sending class)
SMTP helps outgoing E-Mail messages to the predefined SMTP mail server.

(3) Composer(E-Mail Editor class)
Composer helps senders with the mail editing service. Senders are required to input receiver's E-Mail address, subject, and message and select receiver's RSA session key exchange public key.

(4) CipherMail(Main E-Mail Window class)
This class is main application window for managing a E-Mail message list and displaying the contents of E-Mail message. And this class performs encryption, decryption, and certification of delivery when the Composer class requests.

(5) KeyManager(Key generation and Public key management class)
Generating RSA key pairs, one for digital signature and the other for session key exchange, and managing public keys of legitimate receivers.

For these operations, create method, getPublicKey method, getPrivateKey method are as follows;

```
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 Implementation Results

5.2.1 Message Editing and Sending
E-Mail message can be sent with three kinds of mode: mode 1) normal E-Mail transfer, mode 2) E-Mail transfer with digital signature and message encryption, and mode 3) mode 2 plus certification of delivery. Users are required to select one of the three kinds of E-Mail transfer mode. In this case, a user select mode 3, use certification of delivery in <Figure 8>.

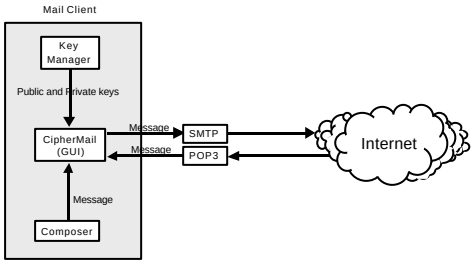

Figure 7. E-Mail System Module.

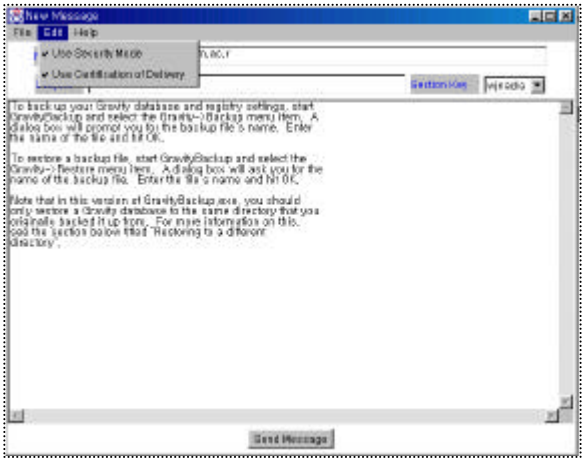

Figure 8. Screen display of Message Composition.

Certification of delivery: AAd3am5hZGlhAAAAg
xuZrsq5+dATONIEZxAiqudx7ppYs+74RARd9H8L6
mjPTuyaDMpm+SOE+pLUI4+BNK2FfwawgzcjdgWG
GrrdBvN7Iz18yDhazbTIYsdvA0MtRxsnB5y7dojk
uKVQf8R093q27yVGW4VAAP/3pIPgYXxJuYLAAAAg
Fur8S7UYEW7aU5qh556yjwvHSmZUiZ3Yxyg2ZNRf
9VKR6Iz22ZBDJAVmEHNNr9L+CYFRgKIm/yxfycwC
…
…

Figure 9. Encrypted Message.

5.2.2 Signature Verification

E-Mail client of the legitimate receiver receives the message and verifies the digital signature. <Figure 10> shows the result of digital signature verification.

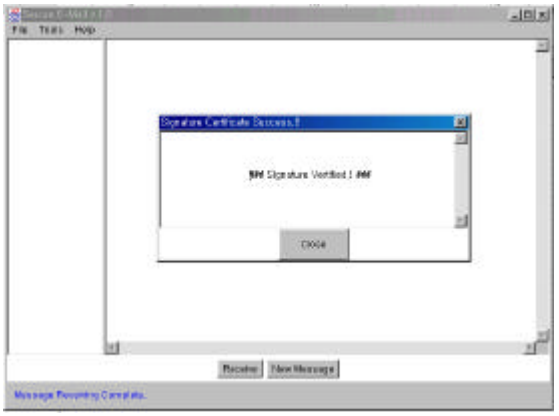

Figure 10. Screen display of Signature Verification of Received Message.

5.2.3 Certification of Delivery

The E-Mail client of the receiver implicitly operates the certification of delivery and the result of this operation goes to the E-Mail client of message origin. The <Figure 11> shows screen display of acknowledgement of delivery from the legitimate receiver's E-Mail client.

5.3 Comparison of Other E-Mail Systems

In this paper, the result of implementation provides users with confidentiality, message integrity, message origin and contents authentication, and non-repudiation. And it provides certification of delivery that is not provided in PGP and PEM. Table 1 shows the comparison of security services of PGP, PEM and proposed E-Mail system.

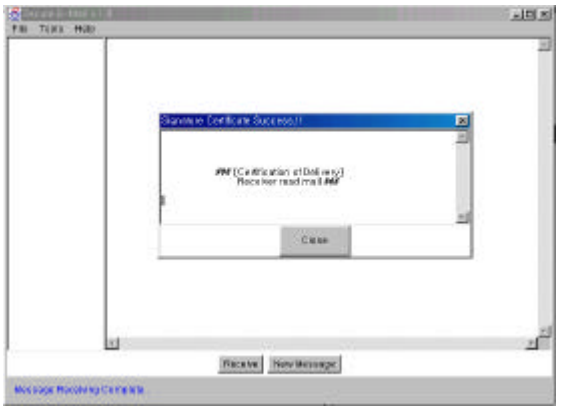

Figure 11. Screen display of Certification of Delivery of Return Message.

Table 1. Comparison of PGP, PEM and Proposed E-Mail System)

Security Service	Proposed System	PGP	PEM
Message Confidentiality	Provided	Provided	Provided
Message Integrity	Provided	Provided	Provided
Origin Authentication	Provided	Provided	Provided
Non-Repudiation	Provided	Provided	Provided
Certification of Delivery	Provided	Not Provided	Not Provided

6. Conclusions

At present, the Internet E-Mail service is exposed to the various types of security attacks. For protecting these security attacks, many kinds of secure E-Mail systems are presented without satisfying user's security requirements. In this paper, a new type of E-Mail system that provides the certification of delivery of E-Mail message is proposed. And this E-Mail system also provides conventional secure E-Mail service such as message encryption, decryption, and digital signature. The proposed model is designed and implemented by Java environments. One of the side effects of this paper is the Java Cryptography package that was developed by the authors. This package provides platform independent solutions with Java developers. Certification of contents of E-Mail message and a fair E-Mail delivery model are some of further study in this paper.

References

Choi, Yongrok & So, Wooyoung & Lee, Yimyoung (1995), *Network Security*, GreenPublish.

Harold, Elliotte Rusty (1997), *Java Network Programing*, O'REILLY.

Hong, Jupoung & Yoon, Yijung & Kim, Daeho (1994. 6), Analysis of E-Mail system Protection Methods, *Journal of Korean Institute of Information Security and Cryptography*, 4(2).

Cho, Hanjin & Kim, Bonghan & Lee, Jaekwang (1998), Design of A Secure E-Mail System , *Hannam University Industrial Technology Research Institute*.

Kang, MyoungHee (1995), Implementation of Security Service for Internet E-Mail System, *Kwangwoon University, Master of Science Thesis*.

Knudesen, Jonathan (1998), Java Cryptography, O'REILLY.

Lee, Jaepong & Lee, Kisoo & Jang, ChunSeo (1997), Design and Implementation of PGP-based WWW Mail system, *Proceeding of Autumn Conference of Korean Information Science Society*, 24(2), 1997.

Sun Microsystems (1999), *Java 2 SDK, Standard Edition Documentation*.

Oaks, Scott (1998), Java Security, O'REILLY.

Park, Chunsik (1997. 6), Survey of E-Mail systems for Certification of Delivery and Contents, *Journal of Korean Institute of Information Security and Cryptography*, 7(2).

Son, Jinwoo (1999), *Java 2 Programing Bible*, Jongbo Munhwasa.

Schneier, Bruce (1996), *Applied Cryptography*, John Wiley & Sons Inc.

Zhou, J. and Gollmann, D.(1996), A Fair Non-repudiation Protocol, *Proc. of the 1996 IEEE Symposium on Security and Privacy*.

Zhou, J. and Gollmann, D. (1996), Observations on Non-repudiation, *Advances in Cryptology, Proceedings of ASIACRYPT '96, Spring-Verlag*.

임신영

건국대학교 공업화학과 학사
건국대학교 화학공학과 석사
건국대학교 전자계산학과 석사
고려대학교 컴퓨터학과 박사수료
현재: 한국전자통신연구원 전자상거래연구부 전자지불연구팀장
관심분야: 인터넷 보안, 공개키 인증기관, 전자 지불, 홍채인증, 디지털 콘텐츠 정보보호 기술

변옥환

한국 항공대학교 통신공학과 학사
인하대학교 정보공학과 석사
경희대학교 정보통신공학과 박사
현재: 과학기술정보유통센터 슈퍼컴퓨팅인프라개발실
관심분야: distributed computing, Internet traffic engineering, security

함호상

고려대학교 산업공학과 학사
고려대학교 산업공학과 석사
고려대학교 산업공학과 박사
현재: 한국전자통신연구원 전자상거래연구부 부장
관심분야: 디렉토리 시스템, 이동 에이전트, 전자 지불, 전자 화폐

김태윤

고려대학교 산업공학과 학사
미국 Wayne State University 전산과학 석사
미국 Auburn University 전산과학 박사
현재: 고려대학교 컴퓨터학과 교수
관심분야: 전자상거래, EDI, 인터넷 보안, 전자지불, 컴퓨터 그래픽스, 멀티미디어 통신