情報保護學會論文誌
第13券 第2號, 2003. 4

End-to-End 기반의 안전한 웹 메일 시스템 설계

전 철 우*, 이 종 후**, 이 상 호*

Design of a Secure Web-mail System based on End-to-End

Chul-woo Chun* , Jong-hu Lee** , Sang-ho Lee*

요 약

웹 메일 시스템은 이동성이 뛰어나고 누구나 쉽게 사용할 수 있다는 장점으로 인해 차세대 전자우편 시스템으로 주목받고 있다. 그러나 기존의 웹 메일 시스템은 보안 메커니즘이 전혀 적용되고 있지 않거나, 보안 서비스를 제공한다 하더라도 그 강도가 약해 외부의 공격자에게 메일의 내용이 쉽게 노출되거나 위조 및 변조의 위험이 있다. 이와 같은 문제의 해결을 위해 본 논문에서는 전자우편 보안 국제 표준인 S/MIME에 기반한 안전한 웹 메일 시스템을 설계 및 구현하였다. 안전한 웹 메일 시스템은 메일의 송수신 및 저장, 사용자 정보의 저장 등의 기능을 수행하는 서버 부분과 메일 작성 및 읽기, 암호화 및 전자서명 기능을 수행하는 클라이언트 부분으로 구성되어 있다. 기존의 보안 웹 메일 시스템에서는 메일에 대한 암호 연산이 웹메일 서버에서 수행되지만, 본 논문에서 설계 및 구현한 시스템에서는 송신자의 클라이언트 모듈에서 S/MIME에 기반하여 작성한 보안 메일을 서버로 전송하면 서버는 이를 수신자에게 전달하고, 수신자의 클라이언트 모듈은 수신한 메일에 대한 복호화 및 전자서명 확인을 수행한다. 이와 같이 보안 메일의 작성을 비롯한 암호연산을 서버가 아닌 클라이언트 부분에서 처리하도록 함으로써 송신자와 수신자간의 End-to-End 보안을 가능하게 하여 안전성 및 신뢰성을 향상시키고 서버의 부하를 줄여 전체 시스템의 효율을 향상시켰다.

ABSTRACT

Web-mail system is worthy of note as a next generation e-mail system for its mobility and easiness. But many web-mail system does not have any kind of security mechanism. Even if web-mail system provides security services, its degree of strength is too low. Using these web-mail systems, the e-mail is tabbed, modified or forged by attacker easily. To solve these problems, we design and implement secure web-mail system based on the international e-mail security standard S/MIME in this thesis. This secure web-mail system is composed of server system and client system. The server system performs basic mail functions - sending/receiving the mails, storing the mails, and management of user information, etc. And the client system performs cryptographic functions - encryption/decryption of the mails, digital signing and validation, etc. Because client system performs cryptographic functions this secure web-mail system gives its reliability and safety, and provides end-to-end security between mail users. Also, this secure web-mail system increase system efficiency by minimize server load.

Keyword : *secure web-mail, S/MIME, end-to-end security*

Ⅰ. 서 론

인터넷 메일은 여러 가지 인터넷 응용 서비스 가운데에서도 가장 많이 사용되는 서비스로써, 인터넷에 익숙하지 않은 일반인들도 메일 계정을 하나쯤 가지고 있는 것이 보통이다. 인터넷 메일의 사용 초기에는 유닉스(UNIX) 명령어에 익숙한 사용자들 이외의 일반인들은 메일 사용이 매우 어렵고 불편하였으나,

* 충북대학교 컴퓨터과학과(chun605@hanmail.net, shlee@cbucc.chungbuk.ac.kr)
** 충남대학교 컴퓨터과학과(jjongfu@cqcom.com)

POP/IMAP(Post Office Protocol/Internet Messaging Access Protocol) 등의 메일 프로토콜과 전용 메일 클라이언트 프로그램의 개발로 최근에는 일반인들의 메일 사용도 그리 어렵지 않다. 그러나 이와 같은 전용 메일 클라이언트 프로그램 역시 인터넷이나 메일 계정 설정 등에 있어서 컴퓨터에 익숙하지 않은 일반 사용자들에게는 어렵고 불편한 부분이 있으며, 메일을 주로 이용하는 장소가 아닌 다른 장소로 이동하였을 때 사용이 어렵다는 점 등, 여전히 몇 가지 단점이 존재한다.

웹 메일 시스템은 이러한 불편을 해소할 수 있는 차세대 전자우편 시스템으로 각광 받고 있다. 즉, 웹 메일 시스템의 등장으로 인해 일반인들도 쉽게 메일 계정을 소지할 수 있게 되었으며, 웹 브라우저만 작동할 수 있다면 누구나 쉽게 메일 조작이 가능하고 자신이 주로 사용하는 컴퓨터가 아닌 다른 컴퓨터에서도 쉽게 메일을 주고받을 수 있어 이동성이 뛰어나다.

한편, 메일 시스템을 비롯한 인터넷의 발전과 함께 인터넷을 통한 보안 위협은 갈수록 커지고 있다. 이는 최근 잇달아 발생하고 있는 대형 사이트에 대한 해킹 사례를 비롯해 다양한 보안 사고의 발생을 통해 쉽게 알 수 있다.

많은 사용자를 확보하고 있는 메일 시스템 역시 이러한 보안 위협으로부터 예외일 수 없다. 다른 사람의 메일 내용을 도청하거나, 전송 도중에 메일을 가로채 그 내용을 변조하는 공격, 다른 사람으로 위장하여 메일을 발송하는 공격 등의 각종 보안 위협이 메일 시스템에도 상존하고 있다[1,2]. 이와 같은 메일에 대한 보안 위협에 대응하기 위한 보안 메커니즘으로는 PGP(Pretty Good Privacy)와 S/MIME (Secure MIME)이 가장 대표적이다.

그러나 기존의 웹 메일 시스템은 이와 같은 보안 메커니즘이 전혀 적용되고 있지 않거나 보안 서비스를 제공한다 하더라도 그 강도가 매우 약한 실정이다. 이에 따라 웹 메일 시스템은 시간과 장소에 구애받지 않고 메일의 송수신이 가능하며, 전문가가 아니더라도 쉽게 사용할 수 있다는 장점 때문에 널리 사용되고 있음에도 불구하고 중요 문서를 교환하는 용도로는 부적당한 것이 사실이다.

이에 따라 본 논문에서는 안전하면서 동시에 편리한 사용이 가능한 웹 메일 시스템을 설계 및 구현하였다. 본 논문의 2장에서는 S/MIME을 중심으로 메일 보안 메커니즘에 대해서 살펴보고, 메일 시스템에 보안 메커니즘을 적용한 기존 사례에 대해서 분석한다. 3장과 4장에서는 각각 본 논문에서 제안하는 S/MIME을 적용한 안전한 보안 웹 메일 시스템 설계 및 구현 내용에 대해서 살펴보고, 5장에서는 제안 시스템을 평가한다. 마지막으로 6장에서 결론을 맺는다.

Ⅱ. 인터넷 메일 보안

2.1 메일 보안 메커니즘

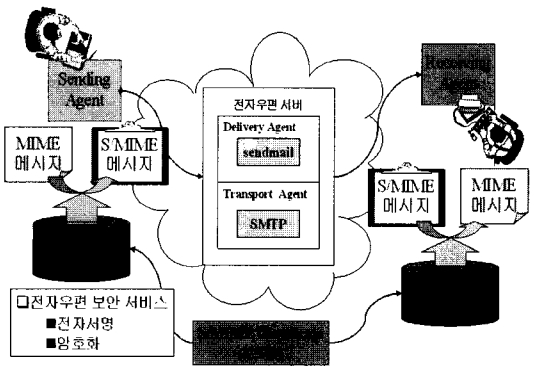
일반적으로 전자우편 보안 서비스는 송신자 인증과 데이터의 기밀성 및 무결성, 그리고 송신 부인 봉쇄 등을 포함한다. 본 절에서는 대표적인 메일 보안 메커니즘인 S/MIME과 PGP에 대해서 살펴보고자 한다.

2.1.1 S/MIME

RSA Data Security Inc.에서 최초 개발된 S/MIME은 현재 S/MIME V3 (version 3) 개발이 완료된 상태이며, 이에 대한 표준화는 IETF(Internet Engineering Task Force)의 S/MIME 작업반(working group)에서 담당하고 있다.

S/MIME은 [그림 1]과 같은 구조를 통해 메일에 대해 전자서명, 암호 등의 보안 서비스를 제공한다. [그림 1]에서 보는 바와 같이 S/MIME은 기존 메일 서비스의 사용자 에이전트(MUA: Mail User Agent)에서 송신하는 메시지에 암호 서비스를 부가시키고 수신 받은 메시지의 암호 서비스를 해석하는데 이용된다[3~7].

즉, 사용자에 의해 작성된 메일 콘텐츠는 일단 MIME



〔그림 1〕 S/MIME 구조

메시지로 변환되며, 이는 다시 PKCS#7에 기반한 CMS (Cryptographic Message Syntax)에 의해서 S/MIME 메시지로 변환된다. CMS에 의해 S/MIME 메시지로 변환되는 과정에서 전자서명, 암호화 등의 보안 서비스가 적용된다. 작성된 S/MIME 메시지는 MUA에 의해서 수신자에게 전송되며, 수신자는 S/MIME 메시지를 CMS에 의해서 MIME 메시지로 변환하여 메일의 내용을 볼 수 있다.

또한 S/MIME에서는 별도의 신뢰구조(Trust model)을 정의하고 있지는 않으나, X.509 인증서를 이용한 키 관리를 전제로 하고있다.

2.1.2 PGP

PGP의 초기 버전은 1990년대 초에 개인에 의해 개발되어 공개 소프트웨어로 사용되었다.

초기의 PGP는 X.509와는 달리 key ring을 이용해 사용자들 상호간에 인증하는 방식을 취하여 키 관리에 있어서 다소간 비효율적이고 복잡한 면이 있었고, MIME을 지원하지 않아 불편한 면이 있었던 것이 사실이다. 그러나 최근에는 IETF에서는 PGP를 MIME과 결합하여 메일 보안에 적합하게 표준화하는 작업(openpgp)을 진행하면서 이러한 문제점들이 해결되어 가고있다.[8,9]. 그러나 S/MIME이 PKI 기반 아래에서 동작하는 반면에 openpgp는 여전히 사용자 자신에 의한 키 관리를 보다 더 강조하고 있다.

2.1.3 S/MIME과 PGP의 비교

앞서 살펴본 S/MIME과 PGP는 대표적인 인터넷 메일 보안 메커니즘으로 다양한 메일 클라이언트 시스템에 적용되어 사용되고 있다.

이 2개의 메커니즘은 암호화 및 전자서명을 통해 메일 메시지에 대해 기밀성, 무결성, 사용자 인증, 부인 봉쇄 등의 보안 서비스를 제공한다는 목적은 동일하지만, 키 관리 및 사용 암호 알고리즘 등에서는 차이점을 갖고 있다. 키 관리와 관련해서는 앞서 설명한 바와 같이 S/MIME은 PKI를 기반으로 하고 있으나, PGP에서는 "Web of Trust"라는 사용자 자신에 의한 키 관리가 강조된다. 두 메커니즘을 비교해 보면 [표 1]과 같다.

S/MIME과 PGP는 모두 IETF를 통해 표준화 작업이 이루어지고 있지만, 현재 보다 널리 사용되고 있는 것은 Outlook/Outlook Express, Netscape Messenger, Eudora, Lotus Notes 등에 적용되어 있는 S/MIME이다. 이와 같이 S/MIME이 국제표준으로 자리잡고 널리 사용되고 있는 이유는 다음과 같이 분석할 수 있다.

〔표 1〕 S/MIME과 openpgp 비교

비교 항목	S/MIME v3	OpenPGP
메시지 형식	CMS 기반	PGP기반
인증서 형식	X.509v3 기반	PGP 인증서
관용 암호 알고리즘	Triple-DES	Triple-DES
전자서명 알고리즘	DSA	DSA
공개키 알고리즘	Diffie-Hellman	ElGamal
해쉬 알고리즘	SHA-1	SHA-1
전자서명을 위한 MIME 타입	multipart/signed 또는 CMS format	ASCII 형식의 multipart/signed
데이터 암호화를 위한 MIME 타입	application/ pkcs7-mime	multipart/ encrypted

우선, 보안 서비스 측면에서 볼 때 S/MIME은 기밀성, 무결성, 사용자 인증, 송신 부인 봉쇄 등 안전한 인터넷 통신에 필요한 보안 서비스를 모두 제공한다. 기밀성은 3-DES, RC2 등 관용 암호기술에 의해 제공되며, 관용 암호기술에 의한 암호화에 사용되는 암호키의 분배는 RSA, Diffie- Hellman 등의 공개키 암호기술을 이용해서 이루어진다. 그리고 RSA, DSA와 같은 전자서명 기술과 SHA-1, MD5 등과 같은 해쉬함수를 통해 사용자 인증, 메시지 무결성, 송신 부인 봉쇄 서비스가 제공된다. 이 때 사용되는 암호 알고리즘들은 이미 안전한 것으로 검증된 알고리즘들이 사용되고 있다.

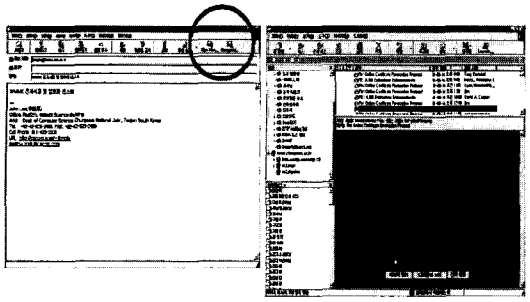
또한 S/MIME은 X.509에 기반한 키 관리 방식을 사용하고 있다. 이 역시 현재 IETF에서 표준화 연구가 이루어지고 있으며, PGP의 키 관리에 비해서는 대규모로 운영이 가능하고 보다 효율적이라고 할 수 있다.

2.2 보안 메일 시스템

2.2.1 S/MIME 기반의 보안 메일 시스템

IETF를 통해서 지속적인 표준화 작업이 진행되고 있는 S/MIME은 현재, Outlook/Outlook Express, Netscape Messenger, Eudora 등 대부분의 메일 전용 클라이언트 프로그램에 적용되어 사용되고 있다.

[그림 2]는 마이크로소프트사의 메일 전용 클라이언트인 Outlook Express에서 S/MIME이 사용된 예이다. 원 안의 버튼을 조작하여 전자서명 및 암호 메시지를 생성할 수 있다. 또한 메시지가 위조 또는 변조 되거나 올바르지 않은 인증서가 사용된 경우 등



〔그림 2〕 Outlook Express에서 S/MIME의 사용

메시지에 보안 문제가 발생했을 경우에는 이에 대한 경고 메시지를 보여준다[8].

S/MIME은 이와 같이 보안성 및 실용성이 뛰어나고 현재 전용 메일 클라이언트 시스템에서는 사용되고 있지만 이동성 및 편리성은 떨어진다.

또한 S/MIME을 이동성과 편리성이 뛰어난 웹 브라우저를 통해 이용 가능한 웹 메일 시스템에 적용한 예는 매우 드물다. 이는 웹 메일 시스템은 본래 SMTP(Simple Mail Transmission Protocol), POP, IMAP 등 메일 프로토콜을 이용하여 메일이 송수신되는 것이 아니라 HTTP 통신을 통해서 메일이 전송되어지기 때문에 메일 보안 메커니즘을 적용하는데 어려움이 있기 때문이다.

따라서 웹 메일 시스템이 전용 메일 클라이언트 시스템을 대신할 메일 시스템으로 떠오르고 있는 가운데 메일의 전송 수단이 HTTP라 하더라도 메일 시스템의 특성을 고려한 메일 보안 메커니즘인 S/MIME이 적용된 보안 웹 메일 시스템의 개발이 반드시 필요하다.

이와 관련해서 기존에 국내에서 웹 메일 시스템에 S/MIME을 적용한 방식을 살펴보면 [그림 3]과 같다. 이 시스템은 일단 송신자와 웹 메일 서버 사이는 뒤에서 살펴볼 SSL 기반의 보안 웹 메일 시스템과 같이 SSL로 보호된다. 즉, 송신자가 작성한 메일 메시지는 S/MIME이 아닌 SSL로 보호되어 서버에게 전송된다. 송신자로부터 메일을 수신한 서버는 SSL로 암호화된 메시지를 복호화한 뒤 다시 S/MIME 메시지로 변환한다. 이 때 서버의 전자서명키를 이용한 전자서명 및 수신자의 공개키를 이용한 메시지 암호화가 수행된다. 서버는 변환된 S/MIME 메시지를 수신자에게 전송한다.

그러나 이와 같은 시스템은 몇가지 문제점을 갖고 있는데 이에 대해서 살펴보면 다음과 같다.

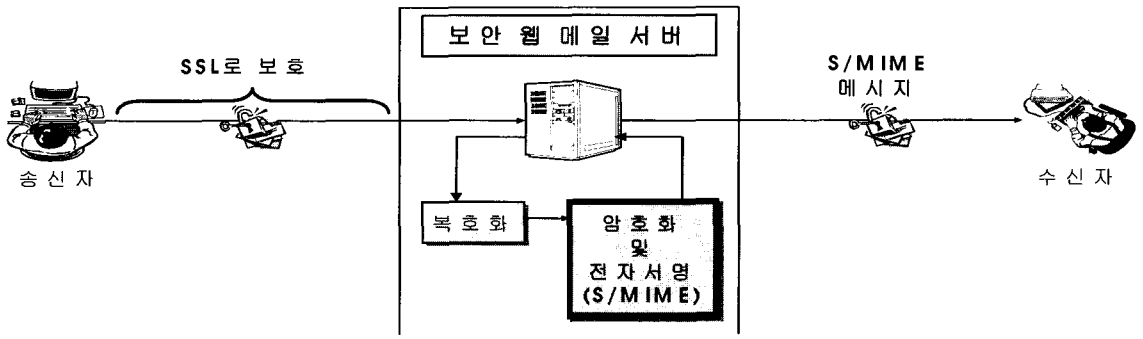
첫째, 송신자가 메일을 작성하여 웹 메일 서버에게 전송할 때 메시지 무결성을 제공받지 못한다. 이는 송신자와 웹 메일 서버 구간에 적용되는 SSL이 데이터에 대한 전자서명을 제공하지 않기 때문에 발생하는 문제이다.

둘째, 웹 메일 서버 관리자에게 메일의 내용이 노출된다. 웹 메일 서버는 송신자로부터 수신한 메일을 일단 복호화한 뒤에 S/MIME을 적용하기 때문에 송신자가 작성한 메일이 웹 메일 서버에게는 노출될 수 밖에 없다.

마지막으로 수신자에게 전송되는 S/MIME 메시지에 전자서명이 적용될 경우, 메일을 작성한 송신자의 전자서명이 아닌 웹 메일 서버의 전자서명이 적용된다. 특히 국내 전자서명법에 의하면 이러한 경우는 메시지의 작성자가 직접 전자서명을 수행한 경우가 아니기 때문에 법의 보호를 받지 못한다.

2.2.2 SSL 기반의 보안 웹 메일 시스템

기존의 보안 웹 메일 시스템은 브라우저와 웹 메일 서버간의 구간을 SSL(Secure Socket Layer)과 같은 웹 보안 메커니즘으로 보호하는 방법을 쓰고있는 경우가 대부분이다. 이는 웹 메일 시스템이 메일 전용 클라이언트와는 달리 HTTP 통신을 한다는 점만



〔그림 3〕 S/MIME 적용 웹 메일 시스템

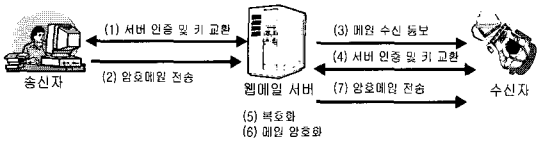
을 고려한 결과이다. SSL 기반의 보안 웹 메일 시스템의 구조는 [그림 4]와 같다[11,12].

송신자는 메일을 전송하기 위해서 웹 메일 서버에 접속하는데, 이 때 SSL 서버 인증서를 이용해서 서버 인증이 이루어지고 송신자 인증은 ID/Password에 의한 인증이 이루어진다. 일단 서버 인증이 완료되며, 서버 인증 과정에서 분배된 암호키에 의해서 송신자와 서버 사이의 데이터가 암호화되기 때문에 송신자가 작성한 메일의 내용은 외부로 노출되지 않는다. 수신자 역시 자신에게 전달된 메일의 내용을 보기 위해서 웹 메일 서버에 접속하고 서버 인증을 거치며, 메일의 내용은 SSL로 보호되는 상태로 수신자의 브라우저에 보여지게 된다.

이와 같은 SSL 기반의 보안 웹 메일 시스템에 대해서 크게 2가지 문제점을 지적할 수 있다.

첫 번째는 보안 메일 서비스가 제공해야 되는 보안 서비스인 사용자 인증, 데이터의 기밀성 및 무결성, 송신 부인 봉쇄 가운데 송신 부인 봉쇄 서비스가 제공되지 않는다는 점이다. 이는 SSL에서 데이터에 대한 암호화만을 제공하고 데이터에 대한 전자서명은 제공하지 않기 때문이다. 이는 일반적으로 웹 통신에 적용되는 SSL의 경우, 클라이언트와 웹 서버 사이에 교환되는 모든 데이터에 대해 전자서명을 제공할 경우에는 전자서명 연산에 소요되는 시간이 너무 오래 걸리기 때문에 매우 비효율적인 통신이 될 수밖에 없기 때문이다. 그러나 메일 서비스에서 송신 부인이 가능할 경우에는 메일의 내용에 따라 심각한 위험을 초래할 수 있다.

SSL 기반 보안 웹 메일 시스템의 두 번째 문제는 웹 메일 서버 관리자가 사용자들의 메일의 내용을 얼마든지 볼 수 있다는 점이다. [그림 2]에서 보는 바와 같이 송신자가 보낸 메일은 일단 웹 메일 서버에서 복호화 된 뒤에 다시 웹 메일 서버와 수신자가 공유하는 키로 암호화되어 수신자에게 전송된다. 따라서 웹 메일 서버에서 메일이 복호화되는 과정에서 평문인 메일이 외부로 누출될 위험이 있으며, 송·수신자 모두와 키를 공유하는 웹 메일 서버의 관리자는 메일의 내용을 얼마든지 볼 수 있다. 또한 메일이



〔그림 4〕 SSL 기반의 보안 웹 메일 시스템

웹 메일 서버에서 보관될 때도, 평문으로 보관되거나 관리자가 알고있는 키로 암호화되어 저장됨으로 웹 메일 서버 관리자는 사용자 메일의 내용을 쉽게 볼 수 있으며, 경우에 따라서는 변조도 가능하다. 즉 SSL 기반의 보안 웹 메일 시스템은 사용자들 간의 end-to-end 보안을 보장할 수 없다.

2.2.3 기타 보안 웹 메일 시스템

지금까지 살펴본 보안 웹 메일 시스템들과는 달리 PGP를 적용한 웹 메일 시스템도 있다[13]. 이 시스템은 사용자 메일에 대한 보안 서비스를 서버에서 처리하지 않고 자바를 이용해서 클라이언트에서 처리하도록 하고 있다.

Ⅱ. 안전한 웹 메일 시스템 설계

이 장에서는 S/MIME 기반의 안전한 웹 메일 시스템의 설계 내용에 대해서 기술한다. 이 시스템은 이동성, 사용의 편리성 등의 웹 메일 시스템의 장점을 살리면서 S/MIME에 기반한 보안 서비스를 제공함으로써 안전성 및 신뢰성 향상을 가져온다.

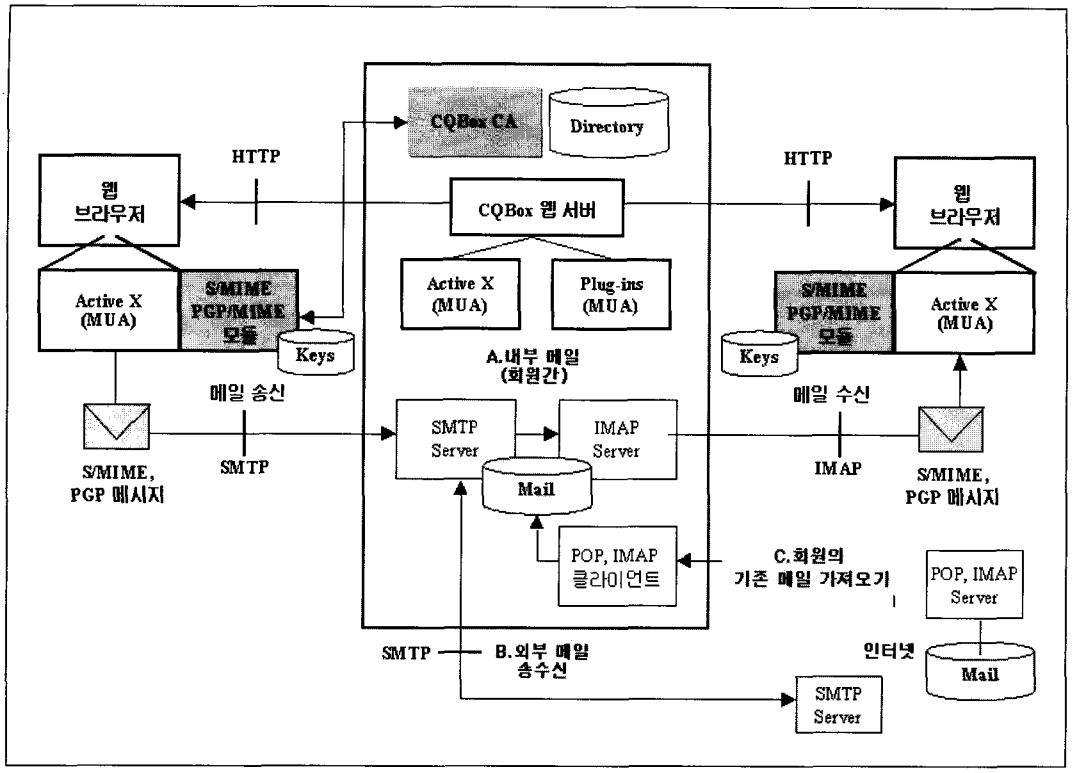
PGP와 S/MIME이라는 메일 보안 메커니즘 가운데 본 논문에서 S/MIME을 선택한 이유는 앞서 기술한 바와 같이 S/MIME이 사실상의 표준으로 좀 더 많은 메일 전용 클라이언트에세 채택되어 사용되고 있으며, 키 관리 등의 면에서 PGP에 비해 좀 더 효율적이기 때문이다.

전체 시스템은 웹 브라우저, 클라이언트 플러그인(plug-in) 모듈, S/MIME 모듈, HTTP 서버, IMAP 서버, CA 서버로 구성되며 S/MIME 모듈을 통해 모든 보안 서비스가 제공된다. 특히, S/MIME 모듈을 웹 메일 서버에 위치하도록 하지 않고 클라이언트 웹 브라우저의 플러그인 모듈 내에 위치하도록 함으로써 SSL 기반의 보안 웹 메일 시스템에서 문제가 되는 end-to-end 보안 및 송신 부인 문제를 해결한다.

3.1 시스템 구성

3.1.1 시스템 구조

안전한 웹 메일 시스템을 통해서 송수신 되는 메일의 형태는 [그림 5]에서 보는 바와 같이 크게 세 가지로 구분할 수 있다. 첫 번째는 내부 회원 사이에



〔그림 5〕 안전한 웹 메일 시스템 구조

서 송수신 되는 메일로, 이 경우에는 모든 보안 기능의 사용이 가능하다. 두 번째는 구현 시스템의 회원이 아닌 사용자와 회원 사이에서 송수신 되는 메일로, 이 경우에는 외부 사용자의 메일 시스템이 S/MIME을 지원할 경우에는 보안 기능의 사용이 가능하지만 그렇지 못할 경우에는 암호화나 전자서명과 같은 보안 서비스가 적용된 메일의 사용이 불가능하다. 마지막으로 내부 회원이 전용 클라이언트를 이용해서 웹 메일 서버에 저장되어 있는 자신의 메일을 가져오는 경우가 있다. 이 경우, 역시 내부 회원이 S/MIME을 지원하는 전용 클라이언트를 사용할 경우에는 암호화나 전자서명 된 메일을 볼 수 있지만 그렇지 못할 경우에는 자신의 메일이라 하더라도 전용 클라이언트를 통해서 메일의 내용을 보는 것은 불가능하다.

또한 구현 시스템은 크게 웹 메일 서버 부분과 웹 메일 클라이언트 부분으로 구성된다. 웹 메일 서버는 메일의 송수신 및 저장, 사용자 정보의 저장 등의 기능을 수행하며 클라이언트 부분은 메일 작성 및 보기 기능, 암호화 및 전자서명과 같은 보안 기능의 수행을 담당한다. 각각의 구성요소별 기능을 살펴보면 다음과 같다.

3.1.2 구성요소별 기능

■ 웹 브라우저

안전한 웹 메일 시스템 이용을 위해서는 사용자 PC에 웹 브라우저가 반드시 설치되어 있어야 한다. 현재 구현 시스템은 Internet Explorer만을 지원한다.

웹 브라우저를 이용해서 웹 메일 서버에 접속한 이후의 메일 송수신, 보안 기능의 사용 등은 모두 웹 브라우저가 아닌 ActiveX로 구현되는 클라이언트 모듈과 보안 웹 메일 서버와의 통신으로 이루어진다.

■ 클라이언트 모듈

메일을 처리하기 위한 클라이언트 모듈로 ActiveX로 구현되어 웹 브라우저에서는 플러그인 형태로 사용된다. 플러그인 모듈은 메일 처리를 위한 MUA(Maile User Agent) 부분과 보안 모듈 부분으로 구성된다. MUA는 메일의 송수신을 처리하는 부분으로 SMTP

를 통한 메일 송신, IMAP을 통한 메일 수신을 처리한다. S/MIME 모듈은 메일의 암호화나 전자서명과 같은 암호 연산을 처리한다.

외부로 송신하는 메일은 우선 S/MIME 모듈을 통해 S/MIME 메시지로 가공된 후 MUA로 전달되어 보안 웹 메일 서버로 전송된다. 수신의 경우에는 MUA가 보안 웹 메일 서버로부터 메일을 수신하여, 이를 S/MIME 모듈로 보내면 S/MIME 모듈에서 원본 메시지로 가공하여 사용자에게 보여주게 된다.

클라이언트 모듈이 웹 브라우저에 플러그인 형태로 설계된 것은 크게 두 가지 이유 때문이다. 첫 째는 이동성을 위해서이다. 전용 응용 프로그램 형태로 구현될 경우에는 보안 웹 메일 시스템을 하나의 시스템에서 계속해서 사용할 경우에는 큰 문제가 되지 않지만 장소를 옮겨가면서 여러 개의 시스템에서 사용할 경우에는 불편함이 생긴다. 두 번째 이유는 서버의 로드를 줄이기 위해서이다. 클라이언트 모듈에서 보안 기능을 처리하지 않고 웹 메일 서버에서 처리할 경우에는 웹 메일 서버가 모든 사용자의 메일을 처리해야 함으로 많은 부하가 생길 수밖에 없다. 이를 클라이언트 모듈에서 처리할 경우에는 사용자 자신의 메일만 처리하면 됨으로 웹 메일 서버의 부하를 줄이고 처리 속도를 향상시킬 수 있다.

■ 보안 모듈

안전한 웹 메일 시스템의 보안기능을 처리하기 위한 보안모듈 부분으로, 앞에서 살펴본 클라이언트 모듈에 포함되어 구현된다. 기존의 보안 웹 메일 시스템의 경우에는 암호화나 전자서명과 같은 암호 연산을 웹 메일 서버에서 수행하도록 하고있다. 그러나 암호연산은 비교적 긴 수행시간을 필요로 하기 때문에 암호연산을 서버에서 처리할 경우 서버의 부담이 매우 커진다. 특히, 동시에 대규모의 사용자가 접속하여 사용하는 웹 메일 시스템의 경우, 이와 같은 서버의 부담으로 인해 서비스 제공시간이 지연될 수 있다. 또한 암호 연산을 웹 메일 서버에서 수행하기 위해서는 사용자의 비밀키가 웹 메일 서버에 보관되어야 하는 문제도 발생한다.

구현 시스템은 이러한 문제를 막기 위해 모든 암호연산을 클라이언트 플러그인 부분에서 처리하도록 하고 있다. 즉, S/MIME 모듈은 처음 접속시 웹 메일 서버로부터 다운로드 되어 클라이언트의 브라우저에서 플러그인 형태로 동작하며, 사용자의 비밀키 및 인증서 관리까지 이 암호모듈에서 처리한다. 그리고 웹 메일 서버는 암호 또는 전자서명 메일을 송신자로부터 수신자에게 중개해주며 복호화나 전자서명 확인은 모두 이 암호모듈에서 처리한다. 이와 같이 웹 메일 서버는 단순히 메일을 중개하고 보관하는 역할만을 하게됨으로 암호연산에 대한 부담을 크게 줄일 수 있으며, 웹 메일 서버 관리자조차 사용자들의 메일 내용을 볼 수 없음으로 안전성 또한 크게 향상된다. 보안 모듈에 대한 보다 자세한 내용은 뒤에서 기술한다.

■ HTTP 서버

사용자와 통신을 위해 웹 메일 서버에 위치한다.

■ IMAP 서버/SMTP 서버

메일 처리를 위해 웹 메일 서버와 동일한 시스템에 위치한다. 즉, 실제적인 사용자 메일의 송수신은 웹 메일 서버 내의 IMAP 서버와 SMTP 서버를 통해서 이루어진다.

■ CA(Certification Authority) 서버

인증서 발행 및 관리 기능을 수행한다. 구현 시스템은 공개키 암호기술을 기반으로 한 S/MIME을 이용하기 때문에 사용자 및 웹 메일 서버 인증서 관리를 위한 CA가 반드시 필요하다. 사용자가 최초로 웹 메일 서버에 등록하게 되면 암호모듈이 브라우저에 설치되고 공개키쌍을 생성한다. 이 때 생성된 공개키는 인증서 신청 양식 (CRS: Certificate Request Statement) 형태로 CA 서버에게 전송되며 CA 서버는 인증서를 발행하여 사용자의 암호모듈과 웹 메일 서버에게 전송한다. 이 CA 서버는 웹 메일 서버와 동일한 위치에 설치될 수도 있고, 그렇지 않을 수도 있다. 이 논문에서 CA 서버는 웹 메일 서버와 동일한 시스템에 위치하며 OpenCA를 수정하여 사용하였다[14].

3.2. 보안 메커니즘 설계

이 절에서는 안전한 웹 메일 시스템에서 보안 서비스를 제공하기 위해서 사용되는 보안 메커니즘의 설계 내용에 대해서 기술한다.

3.2.1 지원 알고리즘

우선 지원하는 알고리즘을 살펴보면 다음과 같이 S/MIME v2 및 S/MIME v3 규격에서 명시된 모든 알고리즘을 지원한다.

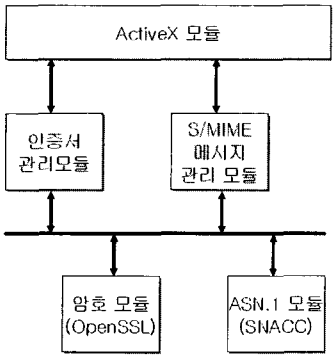
- 해쉬 알고리즘 : SHA-1, MD5
- 전자서명 알고리즘 : RSA, DSA
- 키 교환 알고리즘 : Diffie-Hellman
- 관용 알고리즘 : 3-DES, RC2

3.2.2 S/MIME 모듈

메일 메시지 작성, 메일 송수신 등 메일 기능 외의 보안 서비스는 모두 웹 브라우저 플러그 인 형태로 설계된 S/MIME 모듈에 의해서 제공된다. S/MIME 모듈의 구성은 [그림 6]과 같으며 각 구성 요소의 기능은 다음과 같다.

- ActiveX 모듈 : 사용자 인터페이스 부분이다. 사용자 인터페이스는 Windows 기반의 GUI(Graphic User Interface)로 제공되는데, 이를 통해서 사용자는 메일을 작성하여 송신하거나 수신한 메일의 내용을 볼 수 있다.
- 인증서 관리 모듈 : 수신한 전자서명 메시지의 검증을 위한 인증서 검증 기능과 암호 메시지 송신을 위한 수신인의 인증서 검색 기능을 제공한다.
- S/MIME 메시지 관리 모듈 : S/MIME 메시지의 생성 및 확인 기능을 제공한다. 즉, 전자서명 메시지의 생성 및 검증, 메시지 암호화 및 복호화가 S/MIME 메시지 관리 모듈에서 이루어진다.
- 암호모듈 : 전자서명 생성 및 검증, 암호화 및 복호화 등을 수행하기 위한 암호 라이브러리 부분이다. S/MIME 메시지 관리 모듈 및 인증서 관리 모듈에서 암호 처리는 암호모듈의 함수를 호출하여 사용하는 형태로 이루어진다.
- ASN.1 모듈 : ASN.1 문법으로 표현된 데이터 구조를 C/C++ 형식으로 컴파일 하는 기능을 제공한다.

이와 같은 S/MIME 모듈의 가장 중요한 기능은 S/MIME 메시지의 생성과 확인이다. 이에 대해서 살펴보면 다음과 같다.



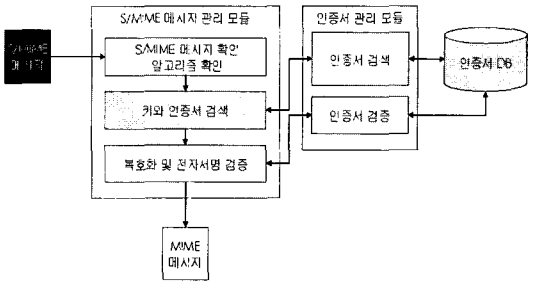
[그림 6] S/MIME 모듈의 구성

우선 S/MIME 모듈 내에서 S/MIME 메시지의 생성은 [그림 7]과 같이 이루어진다.

① S/MIME 메시지 관리 모듈로 MIME 메시지와 생성하고자 하는 S/MIME 메시지의 형태 및 사용 알고리즘이 입력된다. MIME 메시지는 사용자에 의해서 작성된 메일 메시지이며, S/MIME 메시지 형태와 사용 알고리즘은 사용자 인터페이스를 통해 사용자로부터 입력된다. 이 때, S/MIME 메시지 형태는 평문, 전자서명 메시지, 암호 메시지, 전자서명과 암호가 동시에 적용된 메시지 등 네가지 가운데 하나이다.

② MIME 메시지와 사용자 입력으로부터 S/MIME 메시지 형태와 알고리즘을 확인한 뒤에는 전자서명 및 암호화를 위해 인증서 관리 모듈을 통해 키와 인증서 검색을 수행한다. 즉, 전자서명 메시지를 생성하기 위해서는 송신인의 개인키가 필요함으로 S/MIME 보안 모듈 내에 저장된 송신인의 개인키를 로드하고, 암호 메시지 생성을 위해서는 수신인의 공개키가 필요함으로 인증서 DB에서 수신인의 인증서를 검색해서 로드해야 한다.

③ 필요한 키와 인증서가 로드된 뒤에는 전자서명 및 암호화를 수행하여 S/MIME 메시지를 생성한다. 전자서명과 암호화가 모두 적용된 메시지는 먼저 전자서명 메시지를 생성한 후, 이 메시지를 암호화한다. 전자서명 메시지는 원본 메시지를 해쉬값 $H(M)$을 송신자의 개인키 $K_{R\alpha}$로 암호화한 결과와 원본 메시지와 합친 뒤(Concatenation) 이를 압축한 값이 전자서명 값이다. 암호화는 원본 메시지를 압축한 후, 임의로 생성된 세션키를 통해서 이를 암호화한다. 암호화에 사용된 세션키는 송신인의 공개키로 암호화하여 기밀성을 보

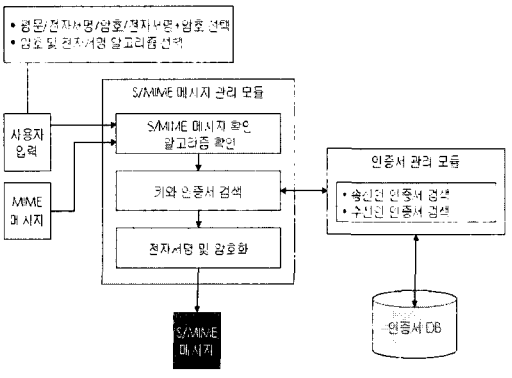


[그림 7] S/MIME 메시지 확인

장한다. 이 후, 이 2개의 값을 합친 것이 암호화의 결과이다. 전자서명과 암호화가 모두 적용되는 경우의 메시지 생성은 앞에서 기술한 바와 같이 전자서명을 먼저 수행한 후 생성된 전자서명값 전체를 암호화한다.

다음으로 S/MIME 보안 모듈 내에서 S/MIME 메시지의 확인은 [그림 8]과 같이 이루어진다.

① 수신한 S/MIME 메시지로부터 메시지의 형태와 사용된 알고리즘을 확인한다.
② 전자서명 및 암호가 적용된 메시지의 전자서명 검증 및 복호화를 위해 키 및 인증서를 로드한다. 전자서명 검증을 위해서는 송신인의 인증서가 필요하기 때문에, 인증서 DB에서 송신인의 인증서를 로드해야 한다. 이 때 S/MIME 메시지에 송신인의 인증서가 첨부되어 함께 전송될 수도 있다. 암호 메시지의 복호화를 위해서는 수신인의 개인키가 필요함으로 S/MIME 보안 모듈 내에 저장된 수신인의 개인키를 로드한다.
③ S/MIME 메시지와 로드한 키 및 인증서를 이용해서 복호화 및 전자서명 검증을 수행한다. 복호화는 자신의 개인키를 이용해 수행하며 전자서명 검증을 위해서는 우선 인증서 관리 모듈에서 인증서 검증이 이루어져야 한다. 인증서 검증에서는 인증경로 상의 모든 인증서를 검증한다. 인증서 관리 모듈에서 인증서 검증이 완료되면 S/MIME 모듈에서는 송신인의 인증서 내에 포함된 송신인의 공개키를 이용해서 전자서명을 확인한다.
전자서명의 확인은 전자서명값의 압축을 풀면 송신인의 개인키로 원본 메시지를 암호화한 값과 원본 메시지가 나오게 된다. 암호화된 값을 수신인의 공개키로 복호화 하면 원본 메시지의 해쉬값이 나온다. 수신인은 수신한 원본 메시지의 해쉬값을 구하여 이를 수신한 해쉬값과 비교하여 동일하면 전송 도중에 메시지가 변조되지 않고 전자서명이 올바르게 수행된 것으로 간주한다. 수신인의 공개키로 암호화된 값이 복호화되지 않거나 비교 결과 해쉬값이 서로 틀리면 전자서명이 올바르지 않거나 전송 도중에 메시지가 변조된 것이다. 암호화된 값의 복호화는 우선 메시지를 암호화 하는데 사용된 세션키를 얻기 위해서 수신인은 자신의 공개키로 암호화된 세션키를 자신의 개인키를 이용해 복호화하는 것으로 시작한다. 이렇게 얻은 세션키로 암호화된 메시지를 복호화한 후, 압축을 풀면 원본 메시지를 얻을 수 있다. 전자서명과 암호화가 모두 적용된 메시지에 대해서는 송신할 때 전자서명 한 값에 대해서 암호화를 수행하였기 때문에 수신할 때는 복호화를 먼저 수행한 후 전자서명을 확인하게 된다.
④ 위의 모든 과정이 완료되면 수신인은 본래의 MIME 메시지를 볼 수 있다.



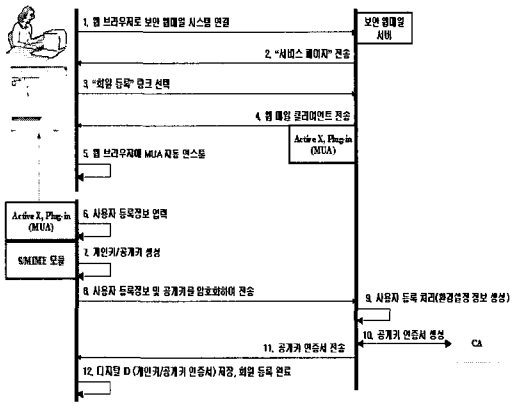
〔그림 8〕 S/MIME 메시지 생성

3.2.3 시스템 동작 절차

이 항에서는 구현 웹 메일 시스템이 동작하는 과정에 대하여 기술한다. 웹 메일 시스템의 동작은 크게 사용자 등록 과정, 메일 송신 과정, 메일 수신 과정으로 구분할 수 있다.

우선 사용자 등록은 [그림 9]와 같이 이루어진다.

① 웹 메일의 사용을 위해서는 웹 메일의 사용자로



〔그림 9〕 사용자 등록 과정

등록을 해야 한다. 이를 위해서 사용자는 웹 브라우저를 이용해 웹 메일 시스템에 연결한다.

② 웹 메일 서버로부터 사용자에게 서비스 페이지가 전송된다.

③ 사용자는 서비스 페이지에서 "회원 등록" 링크를 선택하여 사용자 등록을 시작한다.

④ 사용자는 웹 메일 서버로부터 웹 메일`클라이언트 모듈을 다운로드 받는다. 웹 메일 클라이언트 모듈에는 S/MIME 모듈이 포함되어 있다.

⑤ 다운로드된 웹 메일 클라이언트 모듈을 설치한다. 웹 메일 클라이언트 모듈은 ActiveX로 구현되며 따라서 자동으로 다운로드 및 설치되어 웹 브라우저에서 사용할 수 있다. 여기까지의 과정은 웹 메일 서버와 웹 브라우저의 통신이며, 이후의 과정은 웹 메일 서버와 웹 메일 클라이언트 모듈의 통신이다

⑥ 사용자 등록 정보를 입력한다. 필요한 사용자 등록 정보는 다음과 같다.

- ID
- 한글 이름/ 영문 이름
- 패스워드
- 주민등록번호
- 주소 및 전화번호

⑦ 사용자 등록 정보 입력이 완료되면 사용자가 입력한 패스워드와 S/MIME 모듈 내에서 생성한 임의의 값(random value)을 이용해서 사용자의 개인키와 공개키를 생성한다.

⑧ 사용자 등록 정보와 공개키쌍을 다음과 같은 형태로 웹 메일 서버로 전송한다.

E_{KS} [M || K_{Uuser} || $E_{Password}$ [K_{Ruser}] || $Timestamp$] || $E_{KUserver}$ [KS]

M: 사용자 등록정보
K_{Uuser}: 사용자의 공개키
K_{Ruser}: 사용자의 비밀키
$K_{Userver}$: 웹 메일의 공개키
$Password$: 사용자가 입력한 패스워드
KS: 임의로 생성된 관용키(세션키)

이 때, 사용자 등록정보 M에는 사용자가 입력한 패스워드가 포함되는데 사용자 패스워드는 웹 메일 서버에서도 알지 못하도록 해야 함으로 패스워드를 그대로 전송하지 않고 패스워드의 해쉬값($H(Password)$)을 전송한다. 또한 사용자의 개인키 K_{Ruser} 또한 웹 메일서버가 알지 못하도록 해야하는 정보임으로 사용자만이 아는 사용자의 패스워드를 키로 사용해서 암호화한다.

⑨ 웹 메일 서버는 수신한 사용자 등록 정보를 자신의 DB에 저장한다. 이 때 사용자의 패스워드 대신 패스워드의 해쉬값이 저장되며 사용자의 비밀키 또한 $E_{Password}$ [K_{Ruser}]가 그대로 저장된다. 또한 사용자를 대신해서 CA에게 사용자 공개키에 대한 인증서 발행을 요청한다. CA는 인증서를 발행하여 보안 웹 메일 서버에게 전송한다.

⑩ 웹 메일 서버는 CA로부터 받은 사용자 인증서를 사용자에게 전송한다.

⑪ 사용자가 웹 메일 서버로부터 인증서를 다운로드 받아 자신의 시스템에 설치하면 회원 등록 과정이 완료된다.

다음으로 보안 메일의 송신은 [그림 10]과 같이 이루어진다.

① 메일 송신을 위해 사용자는 우선 웹 메일 서버에 접속해야 한다.

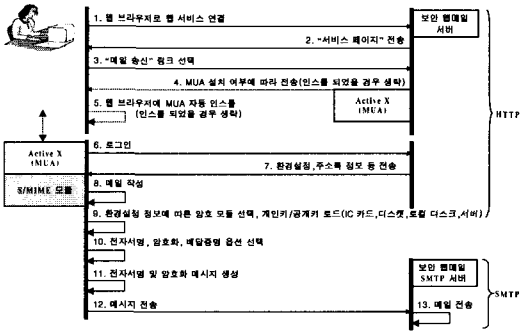
② 웹 메일 서버로부터 서비스 페이지가 전송된다.

③ 사용자는 메일 송신을 위해 '메일 송신' 링크를 선택한다.

④ 웹 메일 서버로부터 웹 메일 클라이언트 모듈이 다운로드 된다. 사용자의 시스템에 웹 메일 클라이언트 모듈이 이미 설치되어 있다면 이 과정은 생략된다.

⑤ 다운로드된 웹 메일 클라이언트 모듈을 설치한다. 이미 설치되어 있다면 이 과정은 생략된다.

⑥ 사용자의 ID와 패스워드를 입력하여 보안 웹 메일 서버에 로그인한다. 이 때 사용자 인증은 사용자 입력한 패스워드가 그대로 웹 메일 서버로

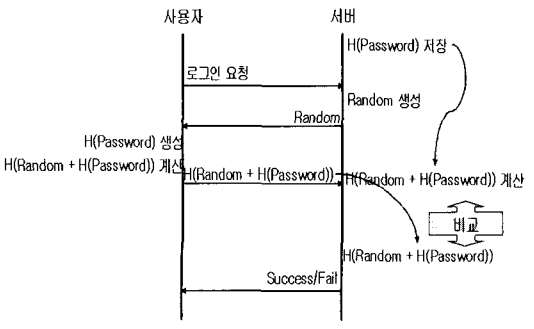


〔그림 10〕 보안 메일 송신 과정

전송되어 웹 메일 서버에 저장되어 있는 패스워드와 비교하는 Basic Authentication 방식이 아닌 Challenge/Response 방식에 의해 [그림 11]과 같이 이루어진다[15].

- 사용자의 패스워드는 사용자의 개인키 보호를 위해 사용되는 정보이므로 웹 메일 서버도 사용자의 패스워드를 알 수 없도록 보호되어야 한다. 따라서 웹 메일 서버에서는 사용자 패스워드의 해쉬 결과인 *H(Password)*를 보관하도록 하였다.
- 사용자가 로그인 요청을 하면 웹 메일 서버는 난수 *Random*을 생성하고, 이를 사용자에게 전송한다.
- 사용자는 자신의 패스워드의 해쉬 결과인 *H (Password)'*와 웹 메일 서버로부터 전송 받은 난수 *Random*을 합친 뒤, 이에 대해 해쉬 연산을 수행하고 그 결과 *(H(Random + H(Password)')*를 웹 메일 서버에게 전송한다.
- 사용자로부터 *H(Random + H(Password)')*를 전송받은 웹 메일 서버는 자신이 저장하고 있던 *H(Password)*를 이용하여 *H(Random + H (Password))*를 계산한 뒤, 이를 사용자가 전송한 값과 비교하여 동일하면 사용자 인증에 성공한 것으로 하고, 다음 단계를 진행한다. 이 때, Random은 매번 로그인 할 때마다 변하는 정보이기 때문에 제3자가 가로채 저장하고 있다가 나중에 다시 사용하는 재전송 공격이 불가능하다.

⑦ 웹 메일 서버로부터 사용자의 개인키 및 인증서, 주소록 정보 등이 전송되어 웹 메일 클라이언트 모듈에 저장된다. 이 때, 개인키 및 인증서 등의 보안 기능과 관련된 정보가 서버로부터 다운로드 되어 S/MIME 모듈에 저장된다.



〔그림 11〕 사용자 인증 과정

⑧ 사용자는 메일을 작성한다. 여기까지의 과정은 사용자의 웹 브라우저와 웹 메일 서버 사이의 HTTP 통신이며, 이 후의 과정은 웹 메일 클라이언트 모듈과 웹 메일 서버 사이의 SMTP 통신이다.

⑨ 사용자의 입력에 의해 암호 알고리즘, 키 크기 등이 결정되고 사용자의 공개키가 로드된다.

⑩ 사용자의 입력에 의해 평문/전자서명 메시지/암호 메시지/전자서명과 암호가 모두 적용된 메시지 가운데 하나가 선택되고 수신 확인 등 기타 옵션 사항이 선택된다.

⑪ 전자서명 및 암호화 메시지를 생성한다.

⑫ 메시지가 웹 메일 서버로 전송된다.

⑬ 웹 메일 서버는 수신한 메시지를 수신인에게 전송한다.

마지막으로 보안 메일의 수신은 [그림 12]와 같이 이루어진다.

① 메일 수신을 위해 사용자는 우선 웹 메일 서버에 접속해야 한다.

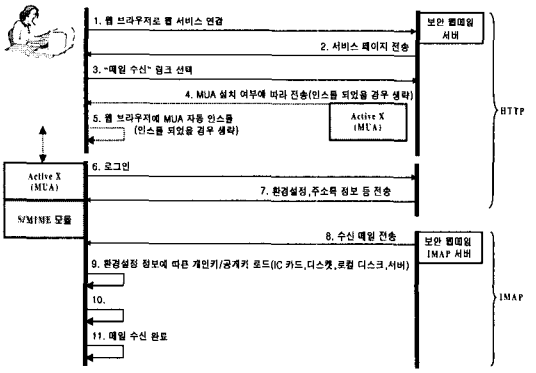
② 웹 메일 서버로부터 서비스 페이지가 전송된다.

③ 사용자는 메일 수신을 위해 '메일 수신' 링크를 선택한다.

④ 웹 메일 서버로부터 웹 메일 클라이언트 모듈이 다운로드 된다. 사용자의 시스템에 웹 메일 클라이언트 모듈이 이미 설치되어 있다면 이 과정은 생략된다.

⑤ 다운로드된 웹 메일 클라이언트 모듈을 설치한다. 이미 설치되어 있다면 이 과정은 생략된다.

⑥ 사용자의 ID와 패스워드를 입력하여 웹 메일 서버에 로그인한다. 사용자 인증 과정은 메일 송신 과정과 동일하다.



〔그림 12〕 보안 메일 수신

⑦ 웹 메일 서버로부터 사용자의 개인키 및 인증서, 주소록 정보 등이 전송되어 웹 메일 클라이언트 모듈에 저장된다. 이 때, 개인키 및 인증서 등의 보안 기능과 관련된 정보는 S/MIME 모듈에 저장된다. 여기까지의 과정은 HTTP 통신이며 이후 과정은 IMAP에 의한 통신이다.
⑧ 웹 메일 서버로부터 메일을 다운로드 받는다.
⑨ 전자서명 검증, 복호화를 위해 필요한 키와 인증서를 로드한다.
⑩ 전자서명 검증, 복호화를 수행한다.
⑪ 원래의 평문 메시지를 보게 되면 메일 수신 과정은 완료된다.

Ⅳ. 안전한 웹 메일 시스템 구현

본 장에서는 앞서 살펴본 설계 내용에 따른 안전한 웹 메일 시스템의 구현 내용에 대해서 기술한다.

4.1 구현 환경

웹 메일 시스템은 [표 2]와 같은 환경에서 구현된다.

웹 메일 서버는 리눅스(Redhat 7.2)에서 구현되었으며, HTTP 통신을 위한 Apache+SSL 서버와 메일 통신을 위한 SMTP, IMAP 서버, 메일의 저장 및 관리를 위한 MySQL 서버가 수정되어 탑재되었다. 서버 개발에 사용된 언어는 C/C++이며 암호 처리는 OpenSSL과 SFL (S/MIME Freeware Library)을 수정하여 사용하였다. 또한 웹 메일 서버의 서비스 페이지 구축에는 PHP를 이용하였다.

웹 메일 클라이언트 모듈은 윈도우 기반의 모든 운영체제에서 Internet Explorer에서 ActiveX 형태로 사용된다. 클라이언트의 S/MIME 모듈 부분은 Visual C++, 사용자 인터페이스 부분은 Delphi와 Visual Basic을 이용하였다. 클라이언트 모듈 역시 암호 라이브러리로는 OpenSSL과 SFL을 수정하여 사용한다.

[표 2] 구현 환경

	서버	클라이언트
운영체제	Linux	Windows95/98/ME/2000/NT
사용언어	C/C++	C/C++, Delphi, Basic
암호 라이브러리	OpenSSL 0.9.5, SFL	OpenSSL 0.9.5, SFL
지원 웹 브라우저	-	Internet Explorer 5.0/5.5/6.0

4.2 구현 내용

■ 클라이언트 모듈의 설치 및 사용자 등록

웹 메일 시스템의 사용을 위해서 최초로 서비스 페이지에 접속하면 아직 사용자의 웹 브라우저에 웹 메일 클라이언트 모듈이 설치되어 있지 않은 상태이다. 따라서 웹 메일 서버로부터 웹 메일 클라이언트 모듈이 웹 브라우저로 다운로드 되어 설치된다. 이 때, 클라이언트는 ActiveX로 구현되었기 때문에 다운로드와 설치는 별도의 다운로드 및 설치 과정 없이 자동으로 이루어진다. 설치가 완료되면, Windows95 /98/ME에서는 'C:/Windows/System' 폴더 하위에, Windows NT/2000/XP의 경우에는 'C:/Windows/System32' 폴더 하위에 각각 다음과 같은 파일이 생성된다. 앞에서 살펴본 바와 같이 클라이언트 모듈은 사용자 인터페이스 부분, S/MIME 모듈 부분, MUA 부분으로 구성되며, 이 외에 클라이언트 모듈의 설치를 관리하는 설치 관리 모듈과 사용자 등록정보를 처리하는 모듈이 필요하다.

- SMailInstaller.ocx : 웹 메일클라이언트 모듈의 설치를 관리하는 파일이다. 웹 메일 서버에 접속할 때마다 서버는 이 파일의 유무를 통해 클라이언트 모듈이 설치되어있는지를 판단한다. 즉, 이 파일이 없거나 파일의 버전이 현재 웹 메일 서버에서 지원하는 클라이언트 모듈의 버전보다 낮을 경우에는 웹 메일 클라이언트 모듈의 설치 작업이 자동으로 시작된다.
- SMailmua.dll : MUA 파일이다. 메일의 송수신과 관련된 모든 작업을 수행한다.
- SMailcm.dll : 암호모듈이다. 암/복호화 및 전자서명, 인증서 검증 등 S/MIME 메시지를 생성 및 확인하기 위해 필요한 모든 암호연산과 관련된 모듈은 이 파일에 포함되어 있다.
- SMailRegisterUser.dll : 사용자 등록을 처리하는 모듈이다.
- SMailMUI : 사용자 인터페이스 모듈이다. 웹 메일 클라이언트 모듈은 웹 브라우저 내에서 동작하는 독립적인 응용 프로그램이라고 할 수 있는데 이를 사용하기 위한 사용자 인터페이스는 모두 이 파일에 포함된다.

[그림 13]은 웹 메일 시스템을 사용하기 위해서 회원으로 등록하는 화면이다. 사용자가 [그림 13]의 필드를 채우고 사용자 등록을 요청하면 웹 메일 클라이언트 모듈에서 공개키쌍 생성이 이루어지고, 웹 메일 서버에 회원으로 등록된다.

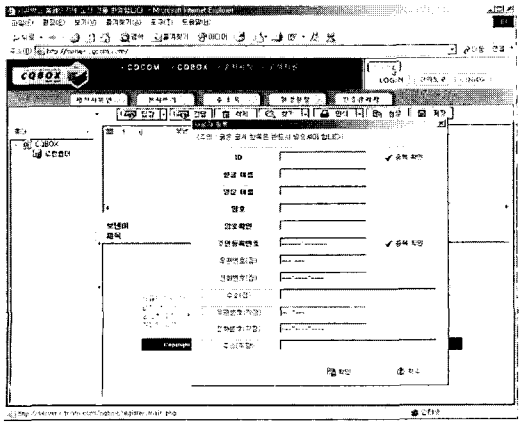
공개키쌍 생성은 사용자가 입력한 패스워드를 이용해서 S/MIME 모듈 내에서 임의로 생성한다. 또한 공개키 인증서는 웹 메일 서버와는 별도로 운영되는 CA 시스템에서 발행하여 클라이언트 모듈로 전송한다. 인증서의 내용은 다음과 같다.

- 버전 : v3
- 일련번호 : 인증서 일련번호
- 소유자 : 사용자명 (메일 어드레스가 포함됨)
- 발행기관 : CA명
- 유효기간 : 인증서 유효기간
- 공개키 알고리즘 : 공개키 알고리즘명(RSA)
- 공개키 : 공개키값(1024비트 RSA 공개키)
- 전자서명 알고리즘 : 전자서명 알고리즘 (sha1WithRSAEncryption)

■ 메일 송신

[그림 14]는 암호화 및 전자서명 된 메일을 작성하는 화면이다. 원 안의 버튼을 클릭하는 것만으로 암호화 및 전자서명이 이루어지게 함으로써 사용자 편의성을 향상시켰다.

전자서명 기능은 수신인의 인증서가 없어도 사용이 가능하지만, 암호화 기능은 반드시 수신인의 인증서가 필요하다. 따라서 암호화 기능을 사용할 때는 로그인할 때 보안 웹 메일 서버로부터 다운로드 된



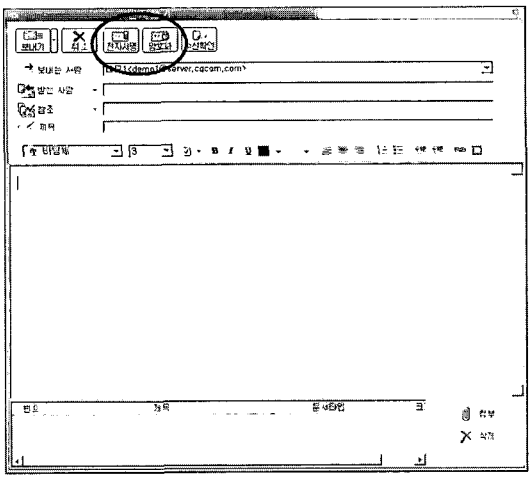
〔그림 13〕 사용자 등록 화면

우에 사용이 가능하다. 만약 수신인의 인증서가 없는 경우에는 수신인의 인증서를 웹 메일 서버로부터 검색해서 다운로드 받아야 한다.

■ 메일 수신

[그림 15]는 암호화 및 전자서명 된 메일을 수신했을 경우의 화면이다. 원 안의 암호화 및 전자서명 메시지를 통해 메일이 암호화되었고, 전자서명 되었음을 확인할 수 있다. 이 때 전자서명 버튼을 누르면 [그림 16]과 같이 송신자의 인증서 정보를 확인할 수 있다.

전자서명 확인 및 복호화에 사용되는 주요 함수는 앞에서 설명된 함수들을 그대로 사용하며 별도의 사용자 조작 없이 웹 메일 클라이언트 모듈에서 암호



〔그림 14〕 보안 메일 보내기

화 및 전자서명 된 메일을 수신했을 경우에 자동적으로 처리한다. 전자서명이 유효하지 않거나 복호화가 불가능할 경우에는 경고 메시지를 출력한다.

전자서명 확인시 필요한 인증서에 대한 정보는 [그림 16]에서 보는 바와 같이 인증서의 모든 필드의 내용을 볼 수 있고 인증 경로를 확인할 수 있도록 되어 있다. 전자서명 메일 수신의 경우, 인증서는 메시지 내에 포함되어 함께 전송된다.

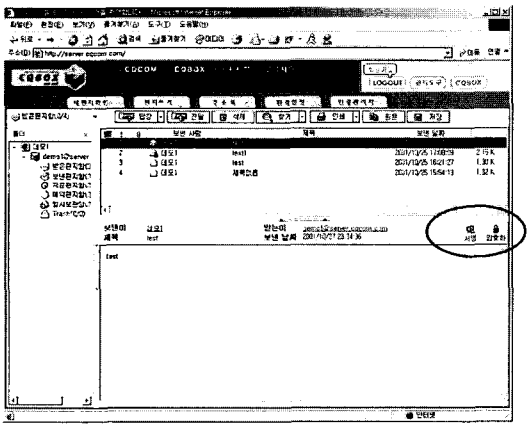
V. 구현 시스템 평가

앞에서 기존의 보안 메일 시스템의 문제점으로 SSL 기반 보안 웹 메일의 취약성에 대하여 기술하였

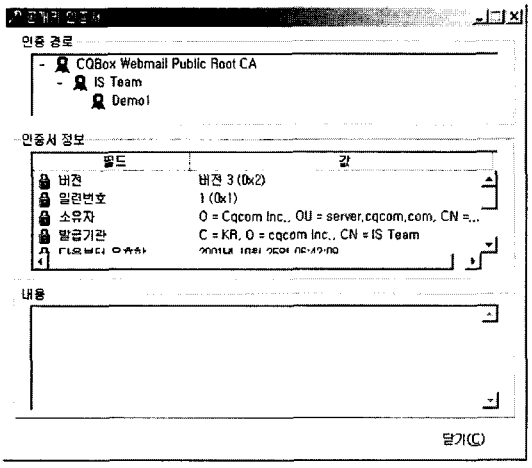
다. 본 절에서는 구현 시스템이 이와 같은 문제점을 어떻게 해결하였는가에 대하여 기술한다.

SSL 기반의 보안 웹 메일 시스템이 갖는 두 가지 문제점인 송신 부인 봉쇄 서비스가 제공되지 않는 점과 관리자가 사용자의 메일 내용을 볼 수 있는 문제점은 S/MIME을 클라이언트 모듈에 적용함으로써 해결이 가능하다.

즉, SSL 기반의 보안 웹 메일 시스템에서는 메일 메시지에 대한 전자서명이 제공되지 않았지만, 이 논문에서 설계 및 구현 웹 메일 시스템에서는 클라이언트가 직접 메일 메시지에 대해 클라이언트의 개인키를 이용해 전자서명이 가능하기 때문에 메시지에 대한 인증이 이루어져 송신 부인이 불가능하다. 또한 전자서명과 마찬가지로 암호화도 클라이언트 모듈에서 수행되기 때문에 송수신 당사자들만이 메일의 내



〔그림 15〕 보안 메일 수신



〔그림 16〕 인증서 정보

사용자 정보에 수신인의 인증서가 포함되어 있을 경우을 볼 수 있고, 웹 서버는 단순히 메일을 중개해 주는 역할만을 하기 때문에 웹 서버 관리자라 하더라도 사용자들의 메일 내용을 알 수 없다. 또한 웹 서버에 저장되는 메일들 역시 사용자의 비밀키를 모르면 그 내용을 볼 수 없도록 암호화되어 저장되기 때문에 웹 서버가 해킹을 당한다 하더라도 사용자들의 메일의 내용이 외부로 노출될 위험이 적다.

추가적으로 웹 메일 서버에 로그인 하는 과정에서도 패스워드의 해쉬값과 난수를 이용하는 Challenge/Response 방식을 응용한 사용자 인증 방식을 사용하여 기존의 ID/Password 방식에 의한 사용자 인증에 비해 패스워드에 대한 도청이나 Brute Force 공격으로부터 좀 더 안전하다. 그리고 웹 메일 시스템은 S/MIME에 기반하기 때문에 S/MIME을 지원하는 모든 메일 프로그램과 호환이 가능하다. 따라서 웹 메일 시스템의 사용자가 아니라 하더라도 S/MIME을 지원하는 메일 프로그램을 사용한다면 웹 메일 사용자와 보안 메일을 주고받는 것이 가능하다. 이와 같은 사항을 요약하면 [표 3]과 같다.

[표 3]에서 보는 바와 같이 웹 메일 시스템은 전용 보안 메일 클라이언트의 장점인 강력한 보안성과 SSL 기반 보안 웹 메일의 장점인 우수한 편리성 및 이동성, 확장성을 함께 제공한다. 웹 메일 시스템의 장점을 살펴보면 다음과 같다.

첫 번째로, 웹 메일 시스템은 많은 시간이 소요되는 암호연산을 클라이언트 모듈에서 처리한다. 즉, 사용자 자신의 메일에 대해서만 전자서명 및 암호화 연산을 수행한다. 따라서 모든 사용자의 메일에 대해서 전자서명 및 암호화 연산을 수행해야 하는 SSL 기반의 보안 웹 메일 시스템에 비해서 시스템 및 서비스의 성능이 뛰어나다. 이 때 클라이언트 모듈의 크기는 약 2M로 처음 접속시에는 다운로드 받는데 다소 시간이 소요되나, 이미 설치된 경우에는 다운로드 받지 않아도 됨으로 설치에는 큰 부담이 없다. 또한 클라이언트에서 암호연산을 수행하는 것이 부담이 될 수 있지만 이는 전용 클라이언트 시스템에서도 동일한 부담으로 작용하며, SSL 기반의 보안 웹 메일 시스템은 서버에서 여러 사용자의 암호 연산을 수행해야 되며 서버와의 통신시간도 고려해야 됨으로 이 역시 큰 부담이 되지 않는다.

두 번째로, 웹 브라우저만 있으면 장소에 관계없이 어디에서든지 사용이 가능하며 조작이 간편하다. 즉, 이동성과 편리성이 우수하다.

〔표 3〕 구현 시스템의 기능 비교

구분 / 기능		구현 웹 메일	전용 클라이언트	SSL 기반 웹 메일
시스템 및 서비스의 성능		사용자 증가에 따른 성능 저하 문제 없음 클라이언트에서 암호연산 수행	사용자 증가에 따른 성능 저하 문제 없음 클라이언트에서 암호연산 수행	사용자 증가에 따른 성능 저하 문제 심각 서버에서 암호연산 수행
편리성 및 이동성		높 음 (최초 접속시 클라이언트 모듈 다운로드 필요)	낮 음	높 음
확 장 성		PGP 등 다른 보안 메커니즘이나 기능의 추가가 용이하고, 다른 응용 서비스로의 확장 또한 용이함	기능의 추가나 다른 응용 서비스로의 확장이 매우 어려움	기능의 추가나 다른 응용 서비스로의 확장이 매우 어려움
보안서비스	기 밀 성	제 공	제 공	제 공
	메 시 지 무 결 성	제 공	제 공	제 공
	송신 부인 봉 쇄	제 공	제 공	제 공 못 함
	사용자인증	강 력 (인증서 또는 OTP 방식)	강력(사용자 PC에서만 사용가능)	취 약(ID/Pass-word 방식)
	키 관 리	X.509 기반	X.509 기반	X.509 기반

세 번째로 전용 클라이언트 방식의 보안 메일 시스템이나 SSL 기반의 보안 웹 메일 시스템에 비해서 새로운 기능을 추가하고 다른 응용 프로그램으로 확장하는 것이 쉽다. 전용 클라이언트 방식의 보안 메일 시스템은 독립적인 응용 프로그램으로 새로운 기능의 추가가 사실상 불가능하며 SSL 기반의 보안 웹 메일 시스템은 단순히 웹 브라우저만을 이용하기 때문에 역시 기능의 추가가 어렵다. 그러나 웹 메일 시스템은 필요한 기능을 구현하여 클라이언트 모듈에 적용하면 되기 때문에 새로운 기능의 추가나 확장이 용이하다.

마지막으로 보안 서비스 측면에서 보면 웹 메일 시스템은 전용 클라이언트 방식의 보안 메일 시스템과 동일한 수준의 안전성을 제공하여, 부인 봉쇄 서비스를 제공하지 못하고 사용자 인증 방식이 취약한 SSL 기반의 보안 웹 메일 시스템에 비해서 뛰어나다. 즉, 웹 메일 시스템은 S/MIME에 기반한 암호화 및 전자서명을 통해 기밀성, 무결성, 사용자 인증, 부인 봉쇄 등의 보안 서비스를 제공하며 이를 통해 메일 시스템의 안전성 및 신뢰성을 향상시킨다. 또한 모든 암호연산을 클라이언트에서 처리하도록 하여 웹 메일 서버에서 사용자의 메일 내용을 보거나 위조 및 변조 할 수 있다는 기존 SSL 기반의 보안 웹 메일 시스템의 문제점을 해결하였다.

Ⅵ. 결 론

인터넷의 발전과 함께 각종 보안사고가 급격하게 증가하고 있는 가운데, 가장 널리 사용되는 인터넷 응용인 메일에 대한 보안 서비스의 적용은 필수적이다. 이에 따라 S/MIME, PGP와 같은 보안 메커니즘이 개발되어 메일 시스템에 보안 서비스를 제공하는데 사용되고 있다.

인터넷 메일은 크게 전용 클라이언트를 사용하는 방식과 웹 메일 방식으로 구분할 수 있는데, 최근에는 편리성과 이동성이라는 장점을 가진 웹 메일 방식이 널리 사용되고 있는 추세이다. 그러나 전용 메일 클라이언트 방식의 메일 시스템 대부분이 S/MIME이나 PGP와 같은 암호기술에 기반한 메일 보안 메커니즘에 의해서 보안 서비스를 제공하는 반면 웹 메일 방식의 메일 시스템은 웹 통신에 적용되는 보안 메커니즘인 SSL에 의해서 보안 서비스를 제공하고 있다.

HTTP 통신에 의한 메일 서비스라고 할 수 있는 웹 메일 시스템에 SSL을 이용해 보안 서비스를 제공할 경우에는 송신 부인 봉쇄를 제공하지 못하고 사용자 인증 과정이 취약해지는 문제가 있다. 또한 SSL 기반의 보안 웹 메일의 경우에는 사용자들 간에 송수신되는 메일이 보안 웹 메일 서버 관리자에게 노출될 위험이 크다.

이와 같이 기존 메일 보안 시스템들은 보안 취약성을 갖고 있어 이에 대한 보완이 필요하다. 이에 따라 이 논문에서는 메일 시스템을 안전하게 운용하기 위해 필요한 메일 보안 메커니즘에 대해 조사/분석하고 기존 메일 보안 시스템들이 갖는 보안 취약성

을 분석한 뒤, 안전한 보안 웹 메일 시스템을 설계 및 구현하였다.

본 논문에서 설계 및 구현한 시스템은 S/MIME을 기반으로 하고 클라이언트 모듈에서 암호 연산을 처리하도록 함으로써, 안전한 인터넷 메일의 운용을 위해 필요한 기밀성, 메시지 무결성, 사용자 인증, 송신 부인 봉쇄 등 보안 서비스를 모두 제공하며, 보안 웹 메일 서버 관리자에게도 메일의 내용을 노출시키지 않는 사용자들 간의 end-to-end 보안을 보장한다. 또한 모든 암호연산을 클라이언트 모듈에서 수행하도록 함으로써 서버와의 통신 시간 및 서버의 암호연산 수행에 따른 부담을 덜 수 있다. 그러나 SSL 기반의 보안 웹 메일 시스템에서는 사용자가 별도로 프로그램을 다운로드 받거나 하지는 않지만, 구현 시스템에서는 최초 접속할 때 클라이언트 모듈의 다운로드가 필요하다.

현재 많은 상용 메일 시스템들이 S/MIME을 구현하여 적용하고 있지만 구현 방법의 차이 등으로 인해 상호호환이 이루어지지 않는 경우가 발생하고 있는데, 상호호환성을 포함하여 S/MIME 구현물들에 대한 객관적이고 효율적인 검증을 수행할 수 있는 방안에 대한 연구가 이루어져야 할 것이다.

참 고 문 헌

[1] B. Schneier, C. Hall, "An Improved E-Mail Security Protocol", 13th Annual Computer Security Applications Conference, ACM Press, 1997. 12.

[2] J. Katz, B. Schneier, "A Chosen Ciphertext Attack against Several E-Mail Encryption Protocols", 9th USENIX Security Symposium, 2000.

[3] R. housley, "Cryptographic Message Syntax", IETF RFC2630, 1999. 6.

[4] B. Ramsdell, "S/MIME Version 3 Message Specification", IETF RFC2633, 1999. 6.

[5] B. Ramsdell, "S/MIME Version 3 Certificate Handling", IETF RFC2632, 1999. 6.

[6] P. Hoffman, "Enhanced Security Service for S/MIME", IETF RFC2634, 1999. 6.

[7] E. Rescorla, "Diffie-Hellman Key Agreement Method", IETF RFC2631.

[8] J. Callas, L. Donnerhacke, H. Finney, P. Thayer, "OpenPGP Message Format", IETF RFC2440, 1998. 11.

[9] M. Elkins, D. Del Torto, R. Levien, T. Roessler, "MIME Security with OpenPGP", IETF RFC3156, 2001. 8.

[10] http://www.microsoft.com/office/outlook/evaluation/security.asp.

[11] http://www. daum.net (https://logins.daum.net/Mail-bin/login.cgi?dummy=2026823433).

[12] http://www.hotmail.com

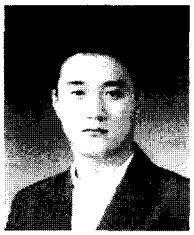
[13] http://www.hushmail.com

[14] http://www.openca.org

[15] W. Simpson, "PPP Challenge Handshake Authentication Protocol (CHAP)", IETF RFC2334, 1996. 8.

〈著者紹介〉

전 철 우 (Chul-woo Chun)
1993년 : 공주영상정보대학 전산분야 강사
2002년 : 충북대학교 전자계산학과 박사
1994년~현재 : 한국전자통신연구원
<관심분야> 컴퓨터 및 네트워크 보안



이 종 후 (Jong-hu Lee)
1997년 : 충남대학교 컴퓨터과학과 졸업
1999년 : 충남대학교 컴퓨터과학과 석사
1999년~현재 : 충남대학교 컴퓨터과학과 박사과정
2000년~현재 : (주) 시큐컴
<관심분야> 컴퓨터 및 네트워크 보안

이 상 호 (Sang-ho Lee)
1976년 : 숭실대학교 전자계산학과 졸업
1981년 : 숭실대학교 전자계산학과 석사
1989년 : 숭실대학교 전자계산학과 박사
1976년~1979년 : 한국전력 전자계산소
1981년~현재 : 충북대학교 전기전자 및 컴퓨터공학부 교수
2001년~현재 : 충북대학교 전산정보원장
<관심분야> Protocol Engineering, Network Security, Network Management, Network Architecture