S/MIME을 사용한 양단간 보안기능 제공하는 SIP 시그널링 기법 구현

현욱, 박선옥, 허미영, 강신각
{whyun, sunok, myhuh, sgkang }@etri.re.kr

Implementation of End-to-End Secure SIP Signaling using S/MIME

Hyun Wook, Huh Mi Young, Park Sun Ok, Kang Shin Gak
Electronics and Telecommunications Research Institute

요 약

RFC3261에 정의된 SIP 프로토콜에는 보안 및 인증과정을 위한 동작 메커니즘을 정의하고 있다. 서비스 제공을 위한 인증이나 권한 점검을 위해서는 HTTP Digest방식이 가장 일반적이고 검증된 방식이긴 하나, 보안의 측면을 보았을 때 안전하지는 않으며, 서버와 단말 또는 단말간의 인증에만 사용되므로 완벽한 형태의 보안을 제공한다고 볼 수 없다. 이러한 보안상의 무결성, 안정성에 있어서의 기능을 충족시키기 위해 RFC3261의 후반부에서는 TLS와 S/MIME의 지원을 권장하고 있으며 현재 IETF에서 진행중인 draft나 각종 RFC에서도 보안 문제를 해결하기 위해 S/MIME과 TLS를 필요한 사항으로 정의하고 있다. 본 고에서는 S/MIME기능을 SIP프로토콜에 적용하기 위해 필요한 사항 등을 구현관점에서 바라볼 것이며, 이를 위해 해당 기능의 구현을 위해 사용된 구현구조 및 관련 API의 설명을 통해 접근하고자 한다.

1 서 론

RFC3261에서는 S/MIME을 사용하여 SIP 메시지의 보안성 및 무결성등에 대한 기능을 보장하는 접근 방식을 채택하고 있다. RFC3261에서 지원하는 보완 관련 기능으로는 HTTP Digest인증 방식과 TLS를 통한 홉간 보안의 보장과 S/MIME을 통한 단말간 보안을 지원하고자 하고 있다. 현재 출시된 SIP 제품들중 TLS나 S/MIME을 통한 보안기능은 시장의 요구가 강하지 않은 관계로 제공되는 경우는 흔치 않다. 그러나 현재 전세계적으로 보안의 중요성이 강조되고 프라이버시의 보호에 대한 요구가 한층 강해짐에 따라 S/MIME을 통한 SIP 단말간 보안기능에 대한 수요가 늘어날 것이다.

본 논문의 2장에서는 RFC3261[1]에 정의된 S/MIME 기능을 사용하기 위한 기본 요구사항을, 3장에서는 S/MIME 기법을 사용하여 SIP 메시지를 암호화/복호화, 서명/인증하는 방법과 해당 기능 구현을 위해 사용된 API 함수, 그리고 동작 구조를 살펴보겠으며, 4장에서 본 논문의 결론을 마무리하겠다.

2 SIP에서의 S/MIME 기능 적용 요구사항

이 장에서는 SIP 메시지에 적용되는 S/MIME 기법이 적용되기 위해 필요한 사항에 대해 다루도록 하겠다.

기본적으로 S/MIME을 적용하기 위해서는 인증서가 필요한데, 인증서는 PK7, PEM을 비롯하여 다양한 포맷으로 존재하며, 본 문서에서 설명하는 API함수들은 PEM파일을 근간으로 하여 구동된다. 인증서의 발급과 관리에 대한 내용은 본 문서에서 논외로 하기로 한다. S/MIME을 적용하고자 하는 사용자는 기본적으로 자신의 Private Key 파일과 Public Key 파일, 그리고 보내고자 하는 상대의 Public Key값을 알고 있어야 한다. S/MIME의 기능을 사용하기 위해 OpenSSL[2]을 사용하였으며, Windows OS에서 개발하였다. 각 S/MIME 기능들에 대해서는 아래의 원본 SIP 메시지에 대한 수행결과를 예제로 하여 설명하도록 하겠다.

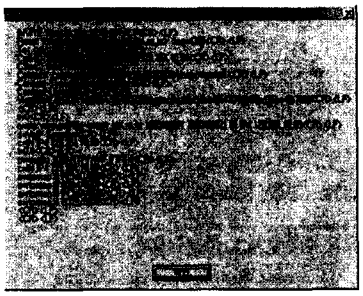

그림 1 General SIP Message

3 S/MIME을 사용한 SIP 메시지 처리

SIP 메시지를 S/MIME 기법을 사용하여 암호화 또는 서명(Sign)하기 위해서는 S/MIME이 적용된 SIP 메시지의 외부에 나타날 부분(Outer Header)과 내부에 표현될 부분(Inner Header)들에 대한 정렬 전처리 과정이 필요하며, 인증/서명 이후의 후처리 과정을 통한 데이터 동기화등의 기능이 필요로 한다. 이 장에서는 각 전처리/후처리 함수에 대한 기본적인 동작에 대한 설명과 각 암호화/서명에 관한 함수의 동작을 설명토록 하겠다.

3.1 S/MIME 적용을 위한 전처리/후처리 함수

S/MIME을 적용하기 이전의 관련 헤더들의 성격에 따라 구분 작업을 해주는 함수와 암호화/서명 이후의 후처리 작업을 수행하는 함수의 동작은 다음과 같다.

3.1.1 SMIME_Preprocess

SIP 메시지로부터 Inner 헤더로 들어갈 헤더와 Outer 헤더로 위치할 헤더를 구분해서 각기 다른 형태의 저장공간에 재정렬 시키는 역할을 한다. 이렇게 Inner의 부분과 Outer의 부분으로 구분된 SIP 메시지를 S/MIME Encrypt함수의 인자로 넘긴다. 송신자 또는 수신자에 대한 정보를 Inner로 위치시켜 중간단에서의 wire-tapping으로 인한 데이터의 유출을 방지할 수 있다.

3.1.2 SMIME_Preprocess4Sign

SIP메시지를 Sign하기에 앞서서 필요한 데이터를 Inner와 Outer 두 부분으로 구분하는 기능을 수행한다. SMIME_Preprocess함수와는 다르게 Content-Type 헤더의 내용만을 Inner로 추가한다. 이후 수신측에서 수신한 메시지의 무결성을 점검하는데 사용될 수 있다.

3.1.3 SMIME_Preprocess4Verify

S/MIME이 적용되어 서명된 SIP 메시지의 Inner부분을 추출해 냄으로써 Verify과정 이후에 유실될 수 있는 데이터의 백업을 유지시킨다. 이 데이터는 이후 Verify과정의 함수가 수행될 때 원 메시지의 복원에 사용된다.

3.1.4 SMIME_Postprocess

S/MIME이 적용된 SIP 메시지의 Body부분을 재파싱의 과정을 통해 Inner헤더의 값을 복원해 낸 후, 최종 SIP 메시지의 내용에 반영을 하여 원 메시지를 실제 복원하는 작업을 수행한다.

3.2 Encrypt/Decrypt

SIP 메시지의 Body에 해당되는 부분을 상대의 Public Key 파일을 사용하여 Encrypt하여 보내며, SIP Header 부분에 대한 내용은 Encrypt하지 않으며, SIP Header중 Content-Type헤더의 내용을 수정하여 Body의 내용이 Encrypt되었음을 알린다. 이를 수신한 측에서는 자신의 Private Key와 Public Key를 활용하여 Decrypt 하여 원본 메시지를 복원하게 된다.

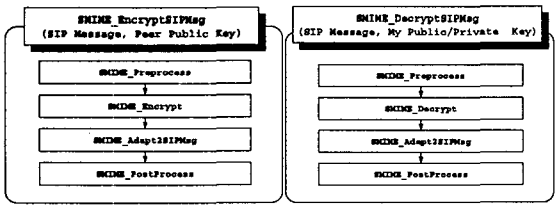

그림 2 SMIME Encryption/Decryption Process

- **SIPMsg_t *SMIME_DecryptSIPMsg(**
  SIPMsg_t *pSIPMsg,
  char *my_public_cert,
  char *my_private_cert)

Encrypt된 pSIPMsg를 같이 입력된 자신의 Public Key와 Private Key를 사용하여 Decrypt하여 원본 메시지를 복원한 이후 결과값을 리턴한다. 실패시 NULL을 리턴한다.

- **SIPMsg_t *SMIME_EncryptSIPMsg(**
  SIPMsg_t *pSIPMsg,
  char *peer_public_cert,
  int algo)

원본 메시지인 pSIPMsg를 같이 입력된 상대의 Public Key 값과 Encryption Algorithm을 사용하여 Encrypt를 수행하며 그 결과값을 리턴한다. 실패시 NULL을 넘겨준다.

아래 그림3은 그림1의 SIP 메시지가 S/MIME이 적용되어 암호화된 내용이다. 그림에서 보는 바와 같이 Content-Type, Content-Disposition, Content-Transfer-Encoding헤더가 추가적으로 생겼으며, 이는 Body부분에 표현될 데이터의 형식을 알려주며, body 부분에는 원 메시지 중 은닉할 필요가 있는 부분은 암호화된 형태로 포함시키고 있다.

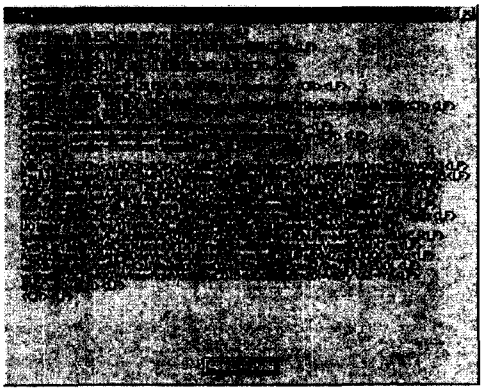

그림 3 Encrypted SIP Message

3.3 Sign/Verify

송신자가 자신이 보내는 SIP 메시지의 무결성을 보장하기 위해 메시지 전체에 대한 Sign을 수행하여 그 결과를 첨부하여 보냄으로써 수신자 측에서의 확인을 통한 메시지의 무결성을 보장받는 방법으로써, 송신자는 보내고자 하는 SIP 메시지 전체를 자신의 Private Key와 Public Key를 사용하여 Sign하여 SIP 메시지의 Body부분에 원본 메시지와 Sign결과값을 첨부하여 보낸다. 수신측에서는 이를 상대의 Public Key를 사용하여 메시지의 무결성을 Verify를 한 이후, Body에 포함되어 있는 원본 메시지를 복원하여 사용한다.

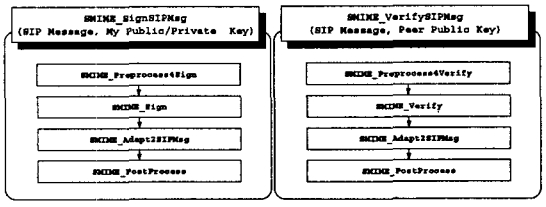

그림 4 SMIME Sign/Verify Process

- **SIPMsg_t *SMIME_SignSIPMsg(**
  SIPMsg_t *pSIPMsg,

```
char *my_public_cert,
char *my_private_cert)
```

원본 메시지인 pSIPMsg를 같이 입력된 자신의 Public Key 값과 Private Key값을 사용하여 Sign를 수행하며 그 결과값을 리턴한다. 실패시 NULL을 리턴한다.

- ```
  SIPMsg_t *SMIME_VerifySIPMsg(
  SIPMsg_t *pSIPMsg,
  char *my_private_cert)
  ```

Sign된 pSIPMsg를 같이 입력된 자신의 Public Key 값을 사용하여 Verify를 수행하며 그 결과값을 리턴한다. 실패시 NULL을 리턴한다.

- ```
  int SMIME_HideSenderInfoSIPMsg(
  SIPMsg_t *pSIPMsg)
  ```

pSIPMsg의 From 헤더 부분의 내용을 변화시키는 함수이며, 실패시 0를 리턴한다.

아래 그림5는 그림1의 메시지가 서명된 결과를 보여준다. 원본 메시지에 서명하여 그 서명결과와 원 메시지를 Body부분에 삽입하며, 수신측에서는 받은 SIP 메시지의 내용을 Verify하여 메시지의 무결성을 점검하게 된다.

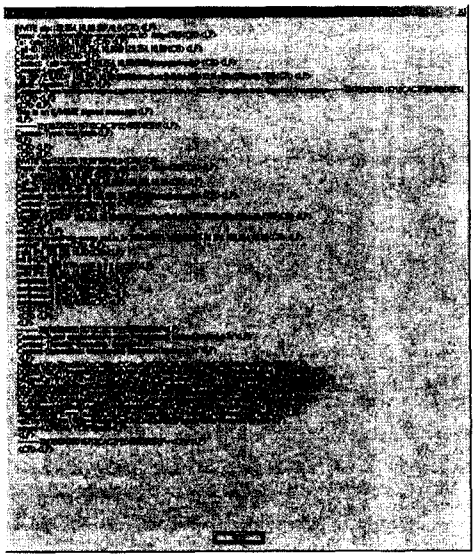

그림 5 Signed SIP Message

3.4 Encrypt+Sign/Verify+Decrypt

이 방식은 위 두가지 방식을 같이 적용함으로써 메시지의 무결성과 보안성을 동시에 적용받고자 할 경우에 사용된다. 먼저 보내고자 하는 원본 SIP 메시지를 Encrypt한 이후, 그 결과에 대해 다시 Sign을 수행하여 송신을 하고, 수신측에서는 수신받은 메시지에 대해 Verify를 수행한 결과에 대해 다시 Decrypt를 수행하여 원본 메시지를 복원하는 형태로 동작을 하게 되며, 경우에 따라 송신자의 주소 정보를 중간 노드들로부터 숨기고자 할 경우에는 From주소를 변경할 수도 있다.

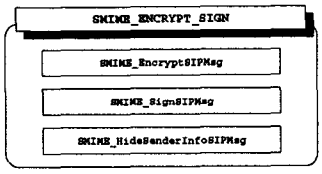

그림 6 SMIME Encrypt and Sign Process

아래 그림 7은 암호화 및 서명이 수행된 SIP 메시지의 내용을 보여준다.

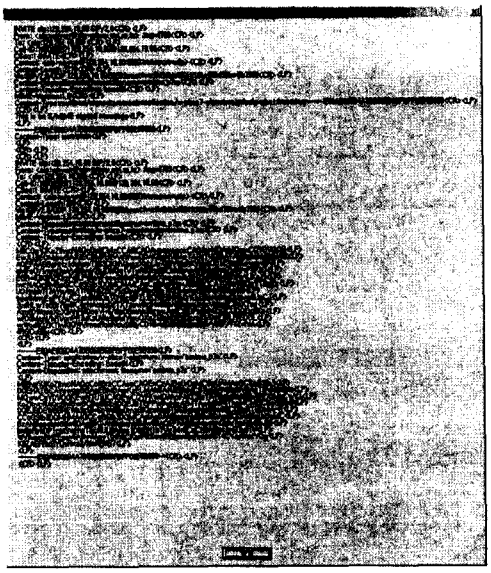

그림 7 SMIME Encrypted and Signed Message

일반적으로 S/MIME이 적용된 메시지는 일반 SIP 메시지보다 크기가 더 크게 되며, 경우에 따라서는 MTU사이즈를 넘어설 수도 있으므로, RFC3261에서는 UDP를 통한 전송보다는 TCP를 통한 전송방식을 선호하고 있으며, 이를 적용하기 위해서는 넘겨받은 메시지의 크기에 따라 자동적으로 TCP로 전환하는 기능이 요구된다.

4 결론

본 논문에서는 RFC3261에 정의된 SIP 메시지에 S/MIME기법의 적용 종류 및 기능구현에 대한 내용을 설명하였다. 앞으로 시장의 성장과 사용자의 보안 및 프라이버시에 대한 요구의 증대로 인해 S/MIME을 통한 메시지 보호기능의 유용성 및 활용성이 극대화될 것이다.

참 고 문 헌

[1] " Session Initiation Protocol," RFC3261
[2] " OpenSSL" , http://www.openssl.org