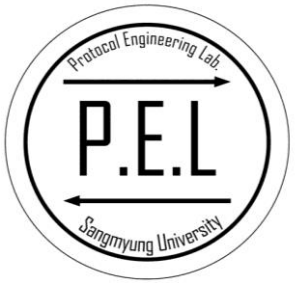


네트워크 보안 에센셜
(7장 전자메일보안)

Sa-rang Wi (sarang@pel.smuc.ac.kr)
Protocol **E**ngineering **L**ab., **Sangmyung** University

Content

- 전자메일보안
- PGP
- S/MIME

전자메일보안

- **전자메일이란?**
  - PC 통신이나 인터넷을 통해 주고 받는 편지
  - 이메일이라고도 함

- **전자메일보안 특징**
  - 단 방향 메시지 보안;일회용 작업
    - ✓ 협상과정, 핸드쉐이크 과정이 없음
    - ✓ 메시지 송신자는 메시지에 사용된 알고리즘 식별자 포함
    - ✓ 메시지 복호화 위한 비밀 키를 수신자의 공개키로 암호화하여 메시지에 포함
  - 전자메일 보안에서 사용하는 상대방의 공개 키를 신뢰하는 방법

전자메일보안

- **전자메일 보안 서비스**
  - PGP
    - ✓ Pretty Good Privacy 의 약자
    - ✓ 1991년 Phil Zimmermann 에 의해 만들어진 전자메일 보안서비스
    - ✓ RFC 2440
  - S/MIME
    - ✓ Secure/Multiple Internet Mail Extension의 약자
    - ✓ 기존의 전자메일 보안 시스템인 PEM(Privacy Enhanced Mail)의 구현의 복잡성, PGP의 낮은 보안성 등 을 보완한 전자메일 보안시스템
    - ✓ RFC 2632~2634

PGP

- **PGP의 기능**
  - 디지털 서명(인증)
    - ✓ SHA-1을 이용해서 메시지 해시코드를 생성
    - ✓ 메시지 해시코드는 송신자의 개인키로 RSA이용 암호화
  - 메시지 암호화(기밀성)
    - ✓ 송신자가 생성한 일회용 세션 키로 CAST-128, IDEA 또는 3DES(64bit CFB)이용해서 메시지 암호화
    - ✓ 수신자의 공개키로 Diffie-Hellman이나 RSA이용해서 세션 키 암호화 후 메시지 첨부
  - 압축
    - ✓ 저장이나 전송을 위해 ZIP알고리즘을 이용하여 메시지를 압축
  - 전자메일 호환성
    - ✓ base-64변환을 이용해 암호화된 메시지를 ASCII문자열로 변환
  - 분할 및 재결합 서비스
    - ✓ 최대 메시지 크기가 정해져 있기 때문에 단편화, 재조합 과정 필요

PGP

- **PGP알고리즘**

공개키 알고리즘(서명, 메시지 암호화)	
ID	Description
1	RSA(encryption or signing)
2	RSA(for encryption only)
3	RSA(for signing only)
16	ElGamal(encryption only)
17	DSS
18	Reserved for elliptic curve
19	Reserved for ECDSA
20	ElGamal(for encryption or signing)
21	Reserved for Diffie-hellman
100-110	Private algorithms

대칭키 알고리즘(암호화)	
ID	Description
0	No Encryption
1	IDEA
2	3DES
3	CAST-128
4	Blowfish
5	SAFER-SK128
6	Reserved for DES/SK
7	Reserved for AES-128
8	Reserved for AES-192
9	Reserved for AES-256
100-110	Private algorithms

해시 알고리즘	
ID	Description
1	MD5
2	SHA-1
3	RIPE-MD/160
4	Reserved for double-width SHA
5	MD2
6	TLGER/192
7	Reserved for HAVAL
100-110	Private algorithms

압축 알고리즘	
ID	Description
0	Uncompressed
1	ZIP
2	ZLIP
100-110	Private methods

PGP

- **PGP 압축**
  - ZIP 압축알고리즘 사용
  - 서명을 수행한 후 암호화 전에 압축
    - ✓ 암호화된 결과를 가지고 평문을 추측하는 행동을 더욱 어렵게 만듦
  - 전자 메일 전송과 파일 저장에 있어 기억 공간을 절약한다는 이점이 있음

PGP

- **PGP 분할 및 재결합 서비스**
  - 전자메일은 보통 최대 메시지 길이 제한 있음 (50,000byte)
  - PGP에서도 50,000byte 이상의 메시지 단편화 하여 전송
  - 단편화 과정은 base-64변환을 포함한 다른 과정 완료 후에 실행
  - 수신측에서는 모든 전자메일 헤더 제거 후 블록 재조립

- **PGP 전자메일 호환성**
  - PGP 결과로 나오는 메시지 블록은 8비트 스트림
  - 기존의 전자메일 시스템에서는 ASCII 문장으로 이루어진 블록만 사용해야 함
  - base-64변환을 사용해 3개의 8bit를 4개의 ASCII 문자로 변환시켜 기존의 전자메일 시스템과의 호환성 문제 해결

PGP

- **PGP에서 사용하는 키**
  - 세션키
    - ✓ 하나의 메시지에 대해서 암호화/복호화할 목적으로만 사용
  - 공개키
  - 개인키
  - 패스워드기반 대칭키
    - ✓ 저장한 개인 키를 암호화하는데 사용

- **키의 필수사항 3가지**
  1. 추측할 수 없는 세션 키 생성 방법
  2. 사용자는 공개키/개인키 쌍을 여러 개 보관 가능
  3. 각 PGP 사용자는 두 가지 파일 필요(상대방의 공개키를 보관하는 파일 + 자신의 공개키/개인키 쌍 보관 파일)

PGP

- **키 식별자**
  - 수신자의 공개키를 이용해 암호화 되어온 세션 키 및 메시지 복호화 필요
  - 사용자들은 여러쌍의 공개키/개인키 쌍을 갖기 때문에 키 식별자를 붙여 공개키들을 구별할 필요 존재

- **키 링**
  - 각 사용자들은 개인키 링와 공개키 링 가짐
    - ✓ 개인키링 : 사용자가 소유한 공개키/개인키쌍들 저장
    - ✓ 공개키링 : 사용자가 알고있는 다른 사용자들의 공개키를 저장

PGP

- **개인키 링**
  - 타임스탬프 : 키쌍이 생성된 날짜/시간
  - 키ID: 공개키의 첫번째(최하위) 64비트
  - 개인키: 패스워드기반 대칭키로 암호화됨
    - ✓ 개인키를 더 안전하게 보관하기 위해 개인키 자체를 키 링에 저장하지 않음
    - ✓ 개인키 링 복구할때 패스워드 입력해야함
  - 사용자ID: 키의 소유자 식별, 보통 사용자 전자메일을 사용

타임스탬프	키ID	공개키	암호화된 개인키	사용자 ID
⋮	⋮	⋮	⋮	⋮
T_i	PU_iand2^{64}	PU_i	$E_{H(P_i)}[PR_i]$	User i

PGP

- **공개키 링**
  - 타임스탬프: 이 항목이 생성된 날짜/시간
  - 키ID: 공개키의 첫번째(최하위) 64비트
  - 소유자 신뢰: 소유자에 대한 신뢰도, 사용자에 의해 지정
  - 사용자ID: 키의 소유자 식별, 보통 사용자 전자메일을 사용
  - 키 합법성: PGP 에 의해서 계산됨
  - 서명에 대한 신뢰: 서명한 사람에 대한 신뢰도, 소유자 신뢰에서 복사됨

타임스탬프	키ID	공개키	소유자 신뢰	사용자 ID	키 합법성	서명	서명에 대한 신뢰
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
T_i	PU_iand2^{64}	PU_i	$trust\_flag_i$	User i	$trust\_flag_i$		

PGP

- **공개키 획득하는 4가지 방법**
  - A가 신뢰할 수 있는 B의 공개키를 얻으려 한다고 가정
    1. B로부터 물리적인 방법으로 키 전달받음
    2. B가 전자메일을 통해 A에게 공개키 전달; A는 공개키 이용 키의 핑거프린트를 만든 후 B에게 전화해서 검증
    3. 양쪽이 모두 믿는 D가 B의 공개키 인증서 서명 후 A에게 전달
    4. CA로부터 B의 공개키 얻음

PGP

- **공개키의 정당성**
  - 사용자가 가진 송신자의 공개키가 정당한지 확인해야 함
    - ✓ 중간자 공격에 의해 바꿔치기 되어있을 가능성 염두
  - PGP에서는 인증기관이 존재하지 않음
  - PGP는 신뢰망 이용
    - ✓ 신뢰망? PGP 사용자가 서로의 공개키에 대해 서로 서명하는 방법

PGP

- **신뢰망**
  - 인증기관을 설치하지 않고 개인끼리 신뢰를 확립!
  - PGP 사용자가 서로의 공개키에 대해 서로 서명하는 방법
    - ✓ 다른 사람의 공개키를 자신이 서명하고 자신이 서명한 사용자의 공개키를 보증하는 형태(공개키 링에 서명필드)
  - 각 사용자가 공개키의 소유자에 대한 소유자 신뢰 값 설정

PGP

- **공개키 링에서의 신뢰망에 사용되는 주요 필드**
  - 소유자 신뢰(Owner_trust)
    - ✓ 사용자가 공개키의 소유자를 신뢰하는 정도를 나타내는 필드
    - ✓ 가질 수 있는 값
      - undefined trust: 모르는 사용자
      - untrusted to sign other keys: 신뢰하지 않음
      - usually trust to sign other keys: 부분 신뢰
      - always trust to sign other keys:항상 신뢰
      - ultimate trust(개인키를 갖는 사용자): 완전히 신뢰

타임스탬프	키ID	공개키	소유자 신뢰	사용자 ID	키 합법성	서명	서명에 대한 신뢰
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
T_i	PU_iand2^{64}	PU_i	$trust\_flag_i$	User i	$trust\_flag_i$		

PGP

- **공개키 링에서의 신뢰망에 사용되는 주요 필드**
  - 서명에 대한 신뢰(Signature_trust)
    - ✓ A의 공개키를 사용하는 사용자들이 서명을 통해 자신의 이름을 걸고 A의 공개키의 진위를 증명해줌
    - ✓ A의 공개키를 가진 사용자는 공개키에 붙는 서명을 붙인 사람에 대한 신뢰도 필요
    - ✓ 그래서, 서명에 대한 신뢰 필드 존재
      - ex) A가 B의 이름이 붙은 키를 받았을 때 이 키에 C의 서명이 붙어있을 경우 소유자 신뢰 필드를 통해 A가 C를 어느 정도 신뢰하는지 확인. 없다면 모르는 사용자라는 값 부여
    - ✓ 가질 수 있는 값
      - undefined trust: 모르는 사용자
      - untrusted to sign other keys: 신뢰하지 않음
      - usually trust to sign other keys: 부분 신뢰
      - always trust to sign other keys:항상 신뢰 이때, 키 합법성 필드 값은 complete trust
      - ultimate trust(개인키를 갖는 사용자): 완전히 신뢰

타임스탬프	키ID	공개키	소유자 신뢰	사용자 ID	키 합법성	서명	서명에 대한 신뢰
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
T_i	PU_iand2^{64}	PU_i	$trust\_flag_i$	User i	$trust\_flag_i$		

PGP

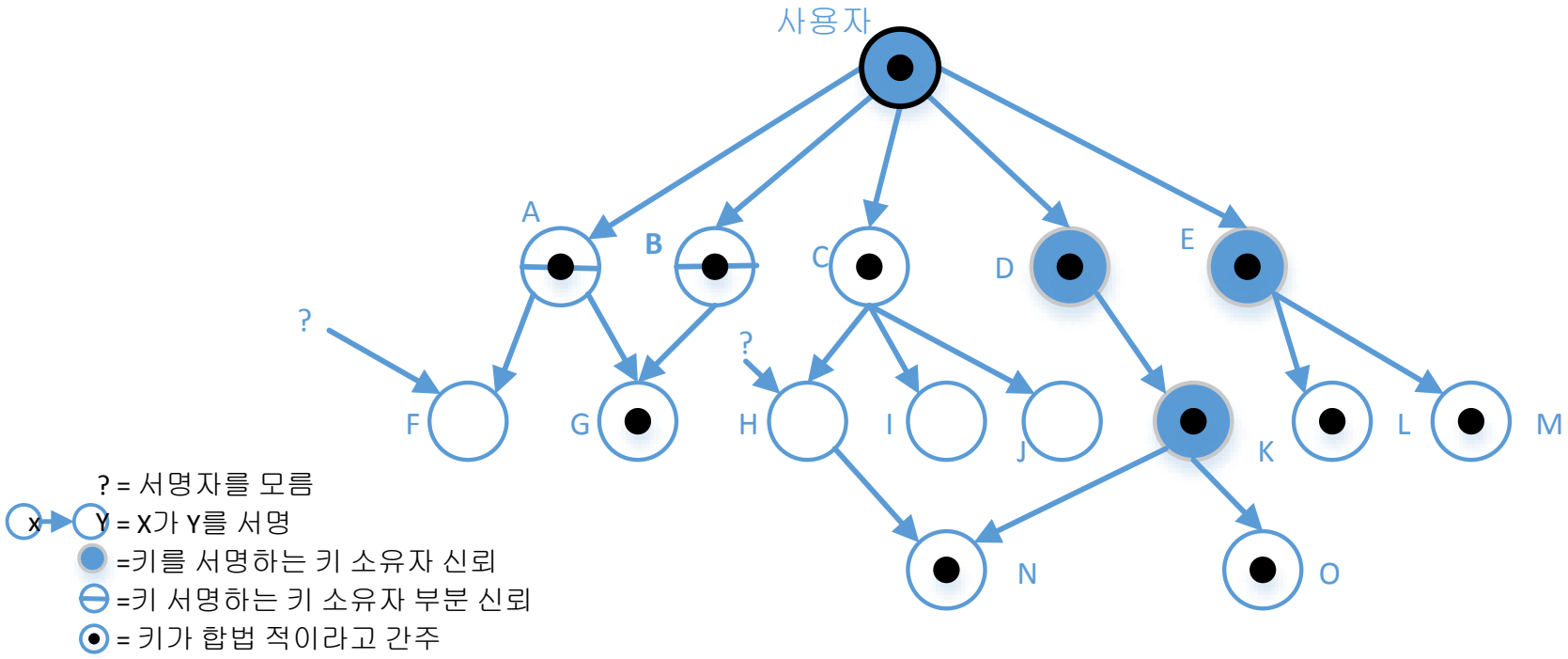
- **공개키 링에서의 신뢰망에 사용되는 주요 필드**
  - 키 합법성(Key_legitimacy)
    - ✓ 사용자가 키를 받았을 때 키를 어느 정도 믿을 수 있는지 나타낸 필드
    - ✓ 값은 서명에 대한 신뢰필드 값을 참고해 PGP가 계산
      - 가중치를 이용해 신뢰 값을 계산

        ex) always trusted일 때는 가중치 1을 부여 usually trusted일 때는 가중치1/2 를 부여
    - ✓ 가질 수 있는 값
      - undefined : 사용자가 알고 있는 사람의 서명이 없을 때
      - untrusted : 사용자가 믿지 않는 사람의 서명이 있을 때
      - marginal trust : 사용자가 어느 정도 믿고 있는 사람의 서명이 있을 때
      - complete trust : 사용자가 완전히 믿는 사람의 서명이 있을 때

타임스탬프	키ID	공개키	소유자 신뢰	사용자 ID	키 합법성	서명	서명에 대한 신뢰
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
T_i	PU_iand2^{64}	PU_i	$trust\_flag_i$	User i	$trust\_flag_i$		

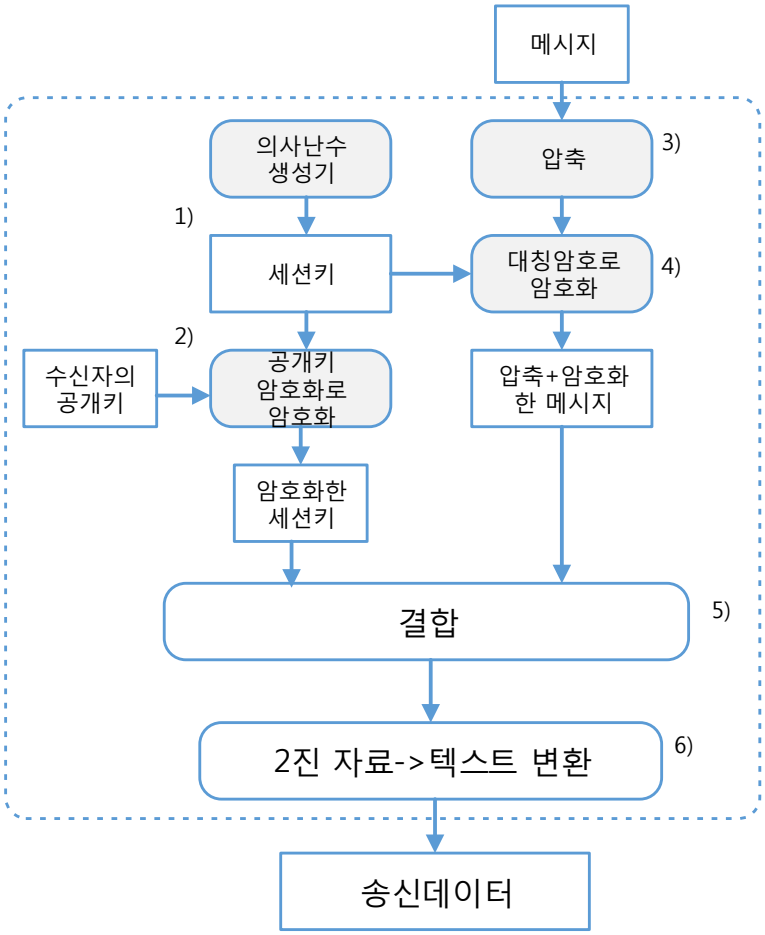
PGP

- **PGP 신뢰망 구성**
  - case 1: 자기 자신의 디지털 서명에 의해 확인
  - case 2: 자신이 항상 신뢰하고 있는 사람의 디지털 서명에 의해 확인
  - case 3: 자신이 부분적으로 신뢰하고 있는 사람들의 디지털 서명에 의해 확인



PGP

- **PGP 암호화**



세션키의 생성과 암호화

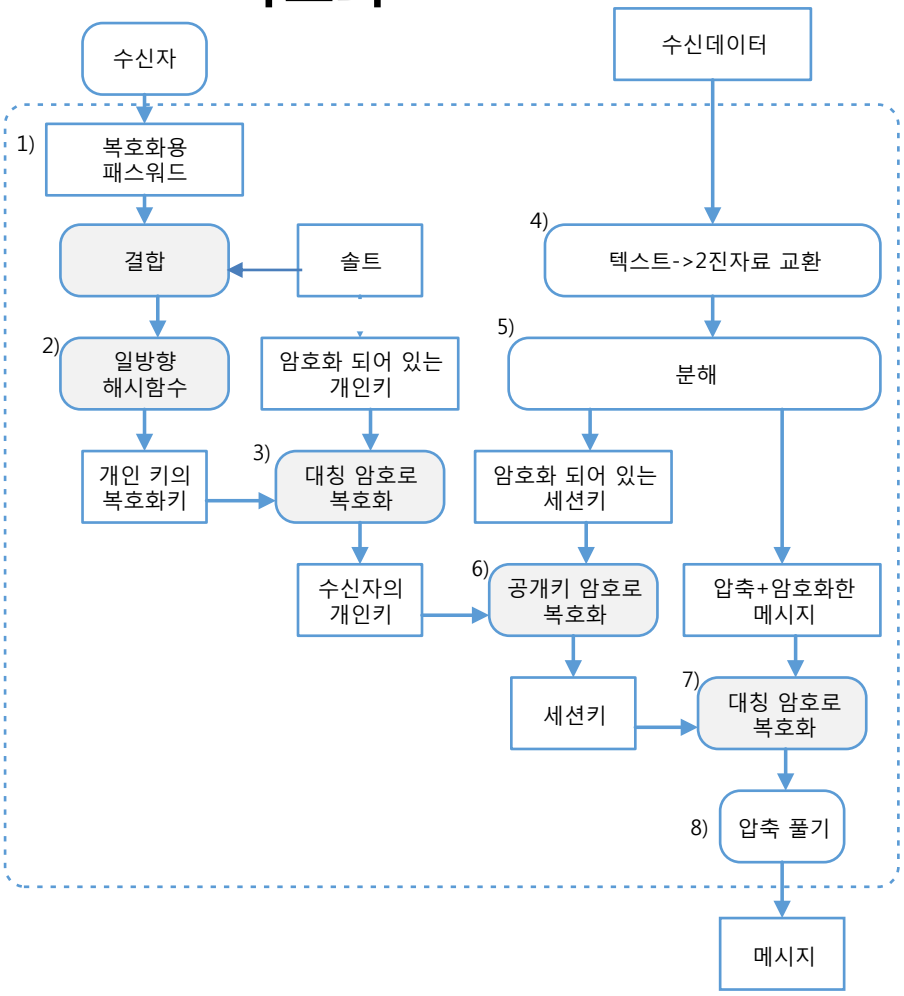
1) 의사 난수 생성기를 사용해 세션키 생성
2) 세션키를 수신자의 공개키 암호로 암호화

메시지 압축과 암호화

3) 메시지 압축
4) 압축한 메시지를 세션키를 이용해 대칭암호로 암호화
5) 암호화한 세션키와 암호화한 메시지 결합
6) 결합한 메시지를 텍스트 데이터로 변환

PGP

- **PGP 복호화**



개인키 복호화

1) 수신자는 복호화를 위해 패스워드를 입력

2) 패스워드의 해시값을 이용해 개인키를 복호화하기 위한 키 생성

3) 키 고리 안에 있는 암호화 되어있는 개인키를 복호화

세션키 복호화

4) 수신데이터를 이진데이터로 변환

5) 2진 데이터를 암호화 되어있는 세션키와 압축+암호화 되어 있는 메시지로 분해

6) 암호화 되어있는 세션키를 공개키 암호로 복호화(3)에서 생성한 수신자의 개인키 사용)

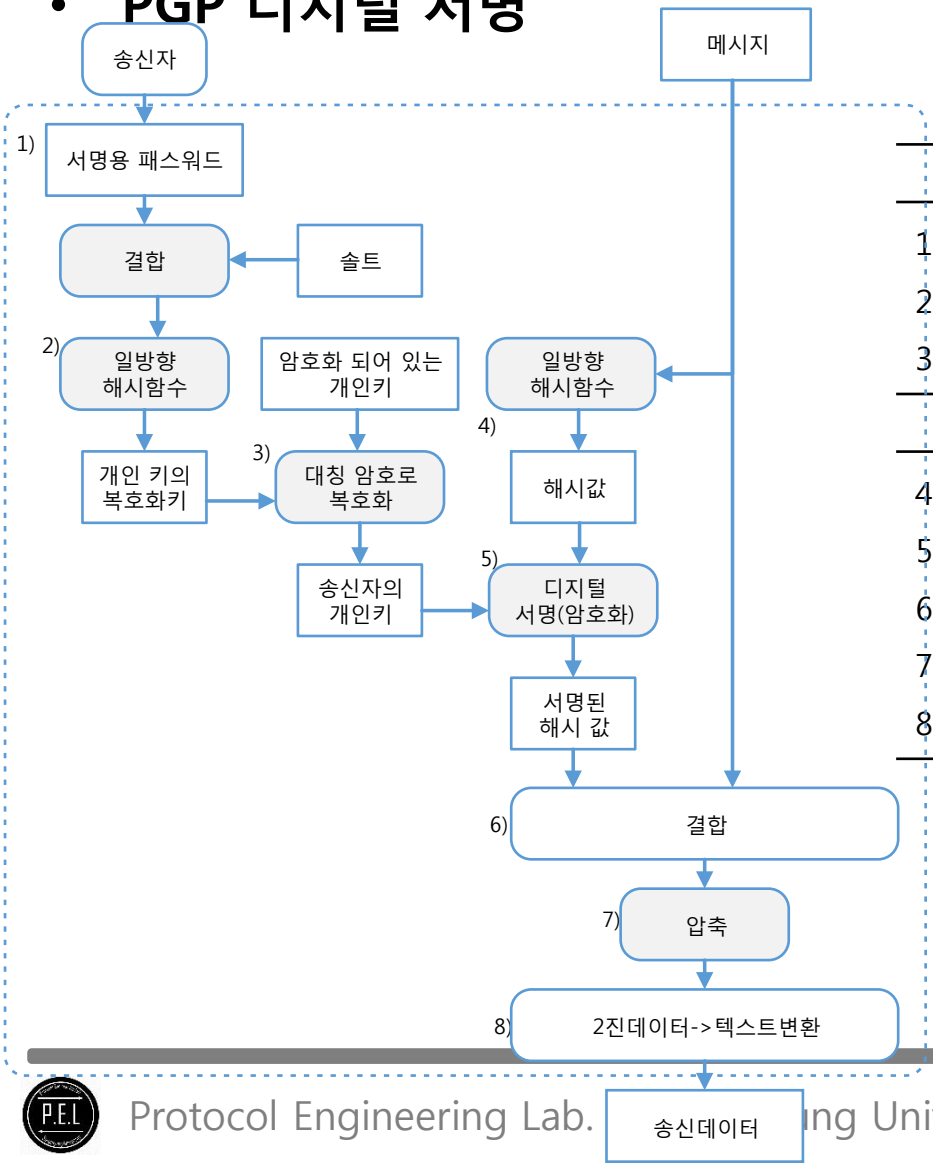
메시지 복호화

7) 압축+암호화 메시지를 대칭암호로 복호화(6)에서 생성한 세션키 사용)

8) 압축되어있는 메시지 풀기

PGP

- **PGP 디지털 서명**



개인키 복호화

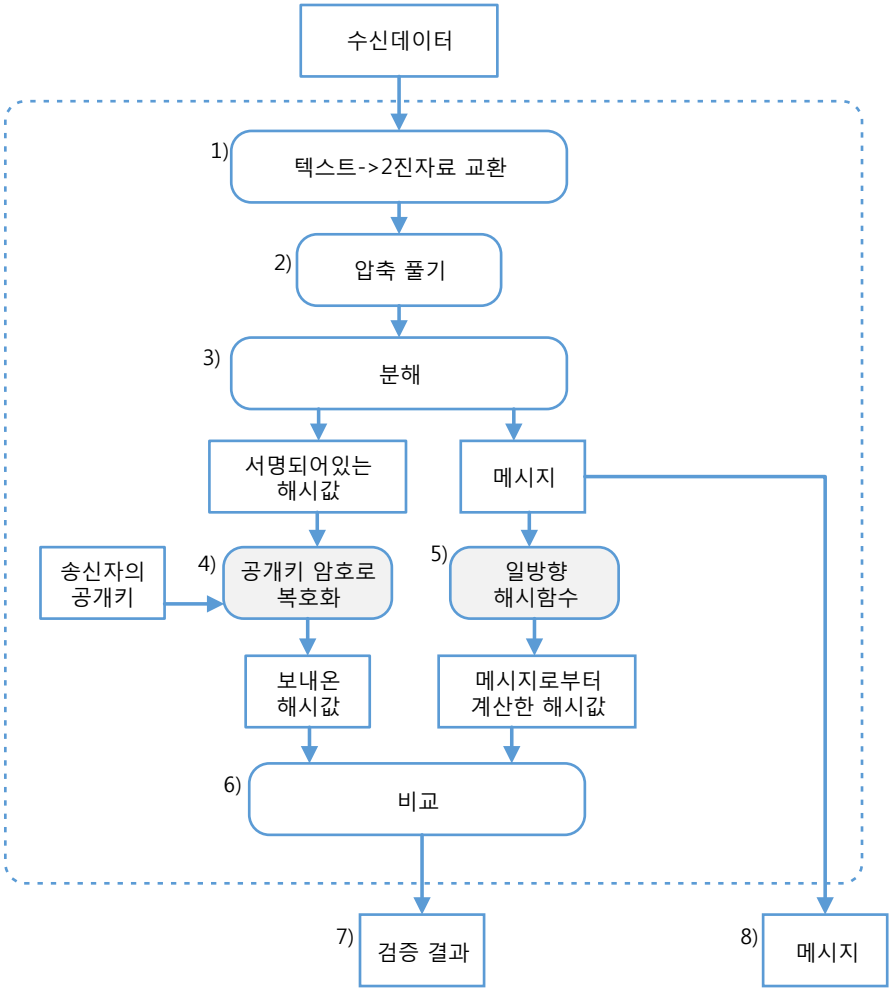
1) 송신자는 서명을 위해 패스워드 입력
2) 패스워드 해시값 취해 개인키를 복호화하기 위한 키 생성
3) 키 고리 안에 있는 암호화 되어있는 개인키 복호화

디지털 서명 작성

4) 메시지 해시값 계산
5) 4)에서 얻은 해시값에 서명(송신자의 개인키 이용)
6) 5)에서 작성한 디지털 서명과 메시지 결합
7) 6)의 결과 압축
8) 7)의 결과 텍스트 데이터로 변환 후 송신데이터가 됨

PGP

- **PGP 디지털 서명 검증**



보내온 해시값의 복원

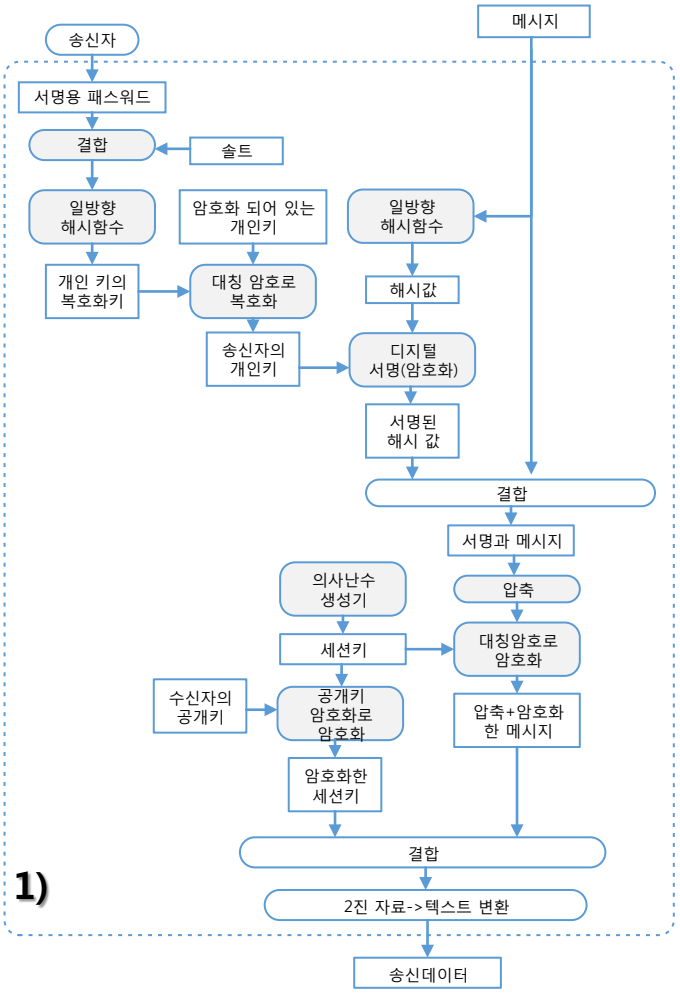
1) 수신 데이터를 이진데이터로 변환
2) 압축된 데이터 품
3) 데이터를 서명되어있는 해시값이랑 메시지로 분리
4) 서명되어있는 해시값을 송신자의 공개키를 사용해 복호화

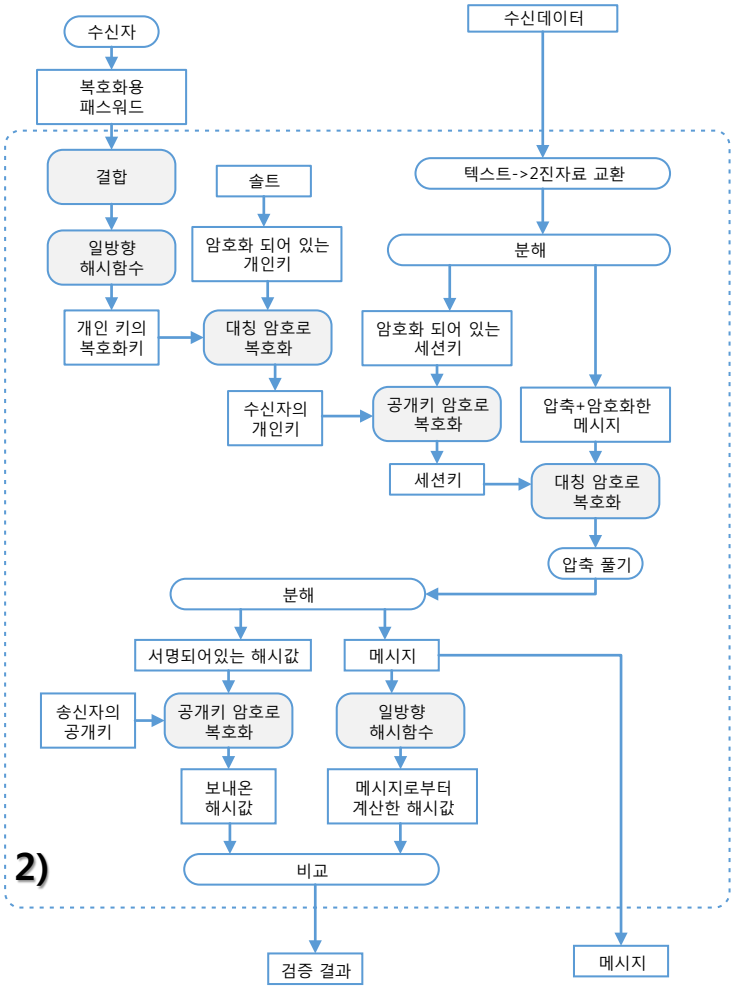
해시값 비교

5) 3)에서 분해한 메시지를 해시함수를 이용해 해시값을 계산
6) 4)에서 얻은 해시값과 5)에서 얻은 해시값 비교
7) 검증의 결과가 같으면 디지털 서명의 검증 성공, 같지 않다면 실패
8) 3)에서 분해한 메시지가 송신자의 메시지임

PGP

- **PGP 전체 과정**

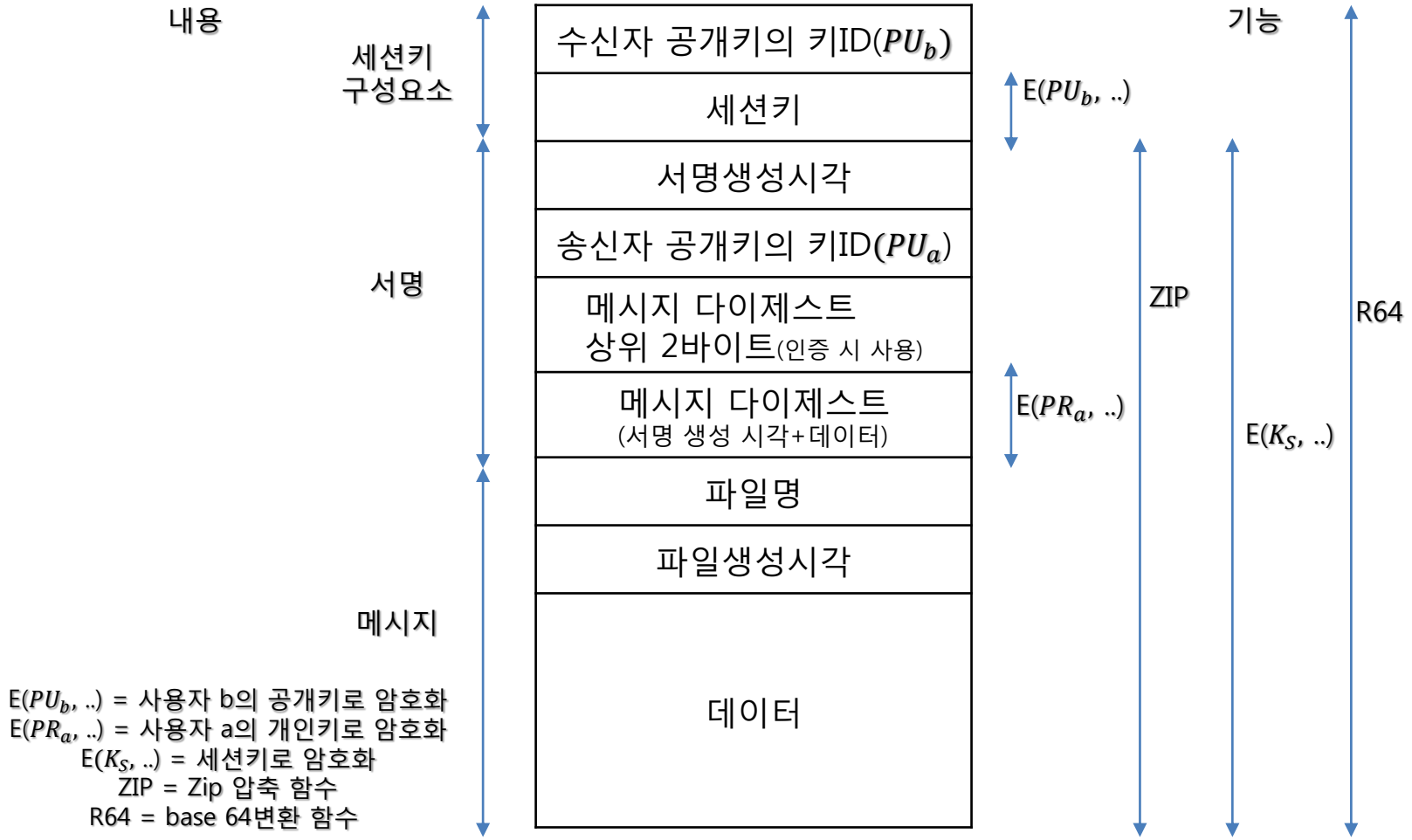




1) 암호화와 서명
2) 복호화와 서명 검증

PGP

- **PGP 메시지 형식(송신자: a , 수신자: b)**



S/MIME

- **S/MIME**
  - 기존의 전자메일 보안 시스템인 PEM(Privacy Enhanced Mail)의 구현의 복잡성, PGP의 낮은 보안성 등 을 보완한 전자메일 보안시스템
  - MIME 데이터에 암호화와 전자서명을 추가한 프로토콜

  - 전자 메일 보안 뿐만 아니라 MIME객체를 전송할 수 있는 모든 프로토콜 보안
    - ✓ HTTP 프로토콜에서 MIME 객체를 전송할 수 있기 때문에 웹 보안에도 사용가능

S/MIME

- **S/MIME 개발 전 메시지 처리 시스템**
  - SMTP(Simple Mail Transfer Protocol)
    - ✓ 전자메일에서의 텍스트 메시지 형식 명세
    - ✓ RFC 822기반
      - ✓ 몇 개의 헤더라인과 그 뒤의 자유로운 형태의 텍스트로 구성

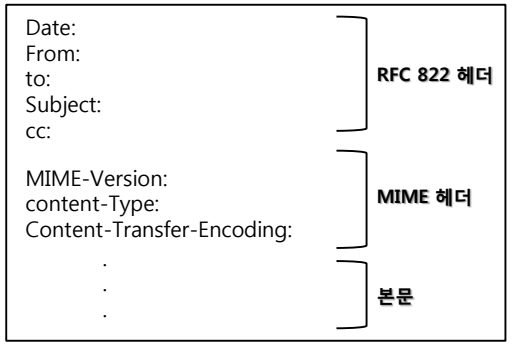
```
Date: Fri,18 Dec 2000 14:19:34 +0900(KST)
From: "Youjin Song" song@mail.dongguk,ac,kr
Subject: The Syntax in RFC 822
to: ofmess@hanmail.net
cc: ns-sice@another-host.com
Hello. This is the syntax in RFC 822
This is a test.
```

  - SMTP/RFC 822 에 기반한 메시지 처리 시스템의 제약 사항
    - ✓ 실행파일이나 이진파일 보낼 수 없음(멀티미디어 데이터)
    - ✓ 8bit코드를 보낼 수 없는데 이것은 2진 바이트 부호를 이용한 한글, 일어, 한자등과 같은 문자 보낼 때 문제가 됨
    - ✓ 특정 크기 이상의 메일 메시지는 거부

⇨ SMTP 를 보안한 MIME 개발

S/MIME

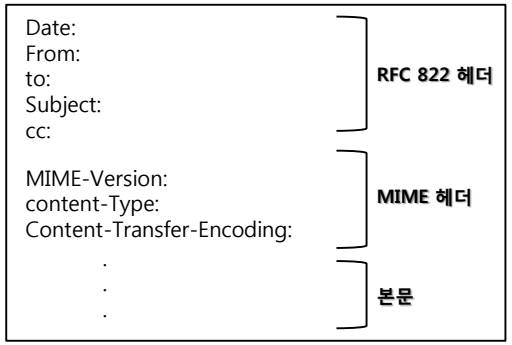
- **MIME**
  - RFC 822 전자메일 시스템과 상호운용 가능하도록 확장된 메시지 포맷



  - MIME-Version
    - ✓ 전자메일메시지가 MIME표준으로 되어있다는 것을 알림
  - Content-Type(메시지 유형)
    - ✓ Type : 메시지 형태(ex) text, audio, image 등 멀티미디어데이터)
    - ✓ Subtype: 메시지 형태에 따른 구체적 내용(ex) text/plaintext, audio/basic, image/jpeg 등)-확장자
    - ✓ 보조정보: type/subtype에 의한 부가적인 정보(ex) text/plain ; charset=iso-8859-*)

S/MIME

- **MIME**
  - RFC 822 전자메일 시스템과 상호운용 가능하도록 확장된 메시지 포맷



  - Content-Transfer-Encoding
    - 멀티미디어 형태를 전송 가능한 형태로 변환
    - RFC1333에서 제안한 base 64 변환 이용

S/MIME

- **S/MIME 보안 서비스**

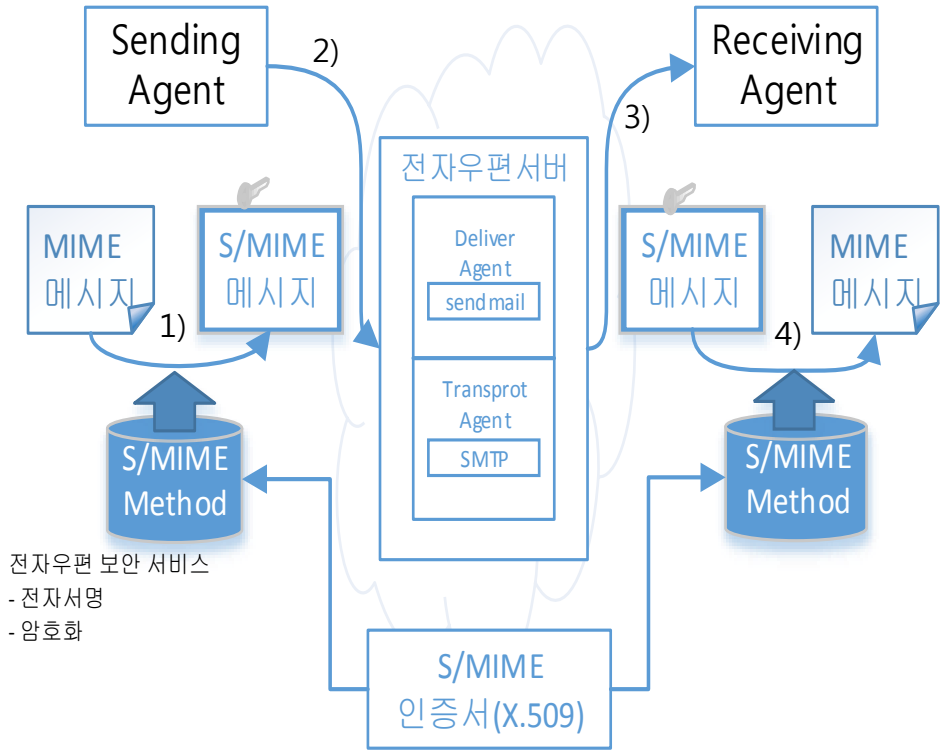
보안서비스	보안메커니즘	암호알고리즘
기밀성	암호화	3DES, Diffie-hellman
무결성	해시함수	SHA-1
인증	인증서	X.509v3
송신부인방지	전자서명	DSA

S/MIME

- **S/MIME 메시지 구성**
  - Enveloped data(봉함된 데이터)
    - ✓ 임의의 타입 데이터의 암호화된 내용과 하나 이상의 다수의 수신자를 위한 암호화된 내용 암호화 키들로 구성
  - Signed data(서명된 데이터)
    - ✓ 디지털 서명은 서명될 내용의 메시지 다이제스트로부터 만들어져 서명자의 개인키로 암호화 서명과 내용은 base64 방식으로 부호화
      - S/MIME 기능을 가진 수신자만 서명된 데이터 메시지 볼 수 있음
  - Clear-signed data(순수한 서명 데이터)
    - ✓ Signed data와 마찬가지로 내용의 디지털 서명이 만들어지나, 디지털 서명만을 base64를 이용하여 부호화
      - S/MIME 기능이 없는 수신자도 메시지 내용은 볼 수 있음
  - signed and enveloped data
    - ✓ 암호화만하는 또는 서명만하는 개체가 중첩되는 경우, 암호화 메시지는 서명을, 서명 데이터는 암호화를 할 수 있음

S/MIME

- **S/MIME의 동작**



1) 수신자에게 보낼 메시지에 대해 전자서명, 암호화, 전자서명 및 암호화를 선택 후 S/MIME Method를 이용해서 S/MIME 메시지 생성
2) 메시지를 보내면 메일클라이언트는 서버에 메일 전송하고
3) 수신자의 메일 서버에 메시지 전송되어 수신자는 S/MIME 클라이언트 이용해 메시지 받음
4) S/MIME 메시지를 서명검증, 복호화 등의 방법을 통해 MIME 으로 변환

S/MIME

- **S/MIME의 동작**
  - 1)과정에서 메시지에 전자서명할 때
    - ✓ 원본 메시지를 MIME 형태로 변환
    - ✓ 전자서명 메시지 생성
      - 해시 알고리즘(SHA,MD5) 이용해 임의의 크기를 갖는 메시지를 20byte의 데이터로 압축
    - ✓ 압축된 데이터를 송신자의 비밀키로 암호화하여 서명
    - ✓ 이진 형태의 전자서명 데이터를 base64로 인코딩
    - ✓ 생성된 전자서명 데이터는 Content-Type:application/x-pkcs7-signature 이용 MIME 형태로 변환

서명된 데이터 의 예)

Content-Type: application/pkcs7-mime; smime-type=signed-data; name=smime.p7m Content-Transfer-Encoding: base64 Content-Disposition: attachment; filename=smime.p7m
567GhIGfHfYT6ghyHhHUujpfyF4f8HHGTrfvhJhjH776tbB9HG4VQbnj7
77n8HHGT9HG4VQpfyF467GhIGfHfYT6rfvbnj756tbBghyHhHUujhJhjH
HUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H7n8HHGghyHh
6YT64V0GhIGfHfQbnj75

S/MIME

- **S/MIME의 동작**
  - 1)과정에서 메시지를 암호화할 때
    - ✓ 원본 메시지를 MIME 형태로 변환
    - ✓ 암호화에 사용될 임의의 세션키 생성
    - ✓ 세션키를 수신자의 공개키로 암호화
    - ✓ 생성된 세션키와 메시지를 암호화
    - ✓ 암호화된 데이터를 base64로 인코딩
    - ✓ 생성된 전자서명 데이터는 Content-Type:application/x-pkcs7-mime 이용 MIME 형태로 변환

암호화된 데이터 의 예)

Content-Type: application/pkcs7-mime; smime-type=enveloped-data; name=smime.p7m Content-Transfer-Encoding: base64 Content-Disposition: attachment; filename=smime.p7m
rfvbnj756tbBghyHhHUujhJHjH77n8HHGT9HG4VQpfyF467GhIGfHfYT6
7n8HHGghyHhHUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H
f8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
0GhIGfHfQbnj756YT64V

감사합니다~