NETWORK SECURITY ESSENTIALS

- 전자메일보안 -

Ki woon Moon

Protocol **E**ngineering **L**ab. **Sangmyung** University

2015. 02. 26

Content

- **PGP (Pretty Good Privacy)**
- **S/MIME**

PGP 개요

PGP 개요

- 1990년경에 필립 짐머만(Philip Zimmermann)이 고안
- 현재도 전 세계에서 널리 사용되고 있는 암호 소프트웨어
- PGP라는 이름은 Pretty Good Privacy(매우 좋은 프라이버시)의 약자

PGP의 주요 기능

- **메시지 암호화**

  CAST, IDEA, 3DES, D-H, RSA 사용

- **디지털 서명**

  SHA-1을 사용하여 메시지의 해쉬코드 생성

  DSS 또는 RSA를 사용하여 메시지 다이제스트 암호화

- **압축**

  저장 및 전송을 위해 ZIP으로 압축

- **전자우편 호환성**

  암호화된 메시지를 기수-64변환을 사용하여 ASCII 문자열로 변환

- **단편화와 재조합**

  최대 메시지 크기제한을 수용하기 위해 분할과 재결합

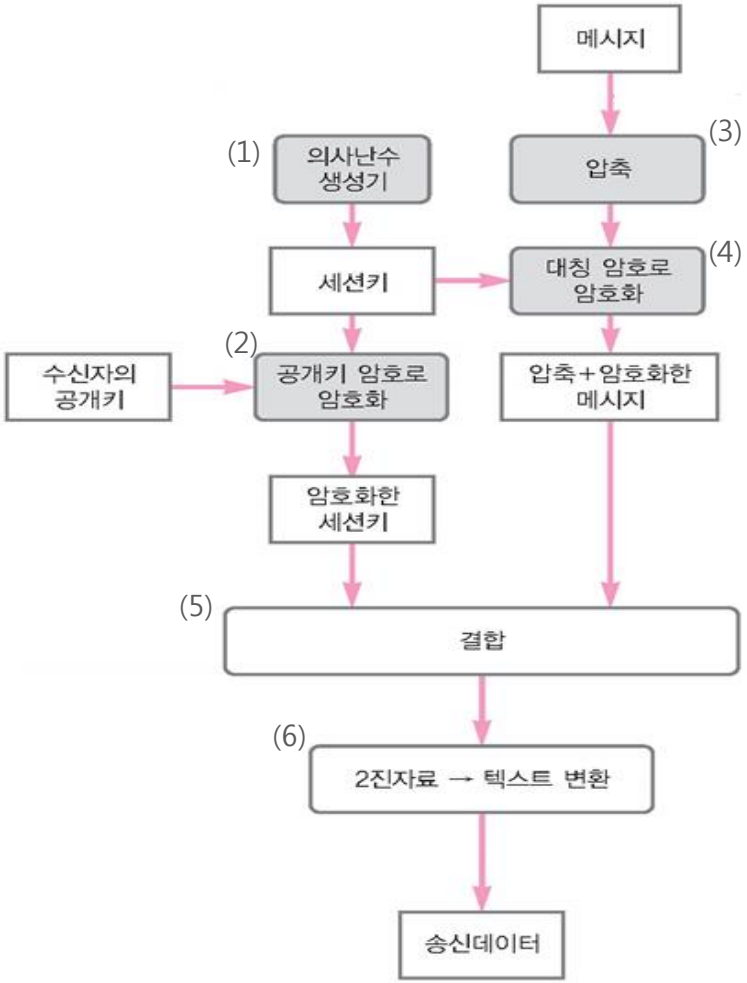
공개 키 알고리즘	
ID	설명
1	RSA (encryption or signing)
2	RSA (for encryption only)
3	RSA (for signing only)
16	ElGamal (encryption only)
17	DSS
18	Reserved for elliptic curve
19	Reserved for ECDSA
20	ElGamal (encryption or signing)
21	Reserved for Diffie-Hellman
100-110	Private algorithms

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

대칭 키 알고리즘	
ID	설명
0	NO Encryption
1	IDEA
2	Triple DES
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

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

PGP에 의한 암호화



(1) 의사난수 생성기를 사용해서 세션 키를 생성

(2) 세션 키를 공개 키 암호로 암호화 (수신자의 공개 키 사용)

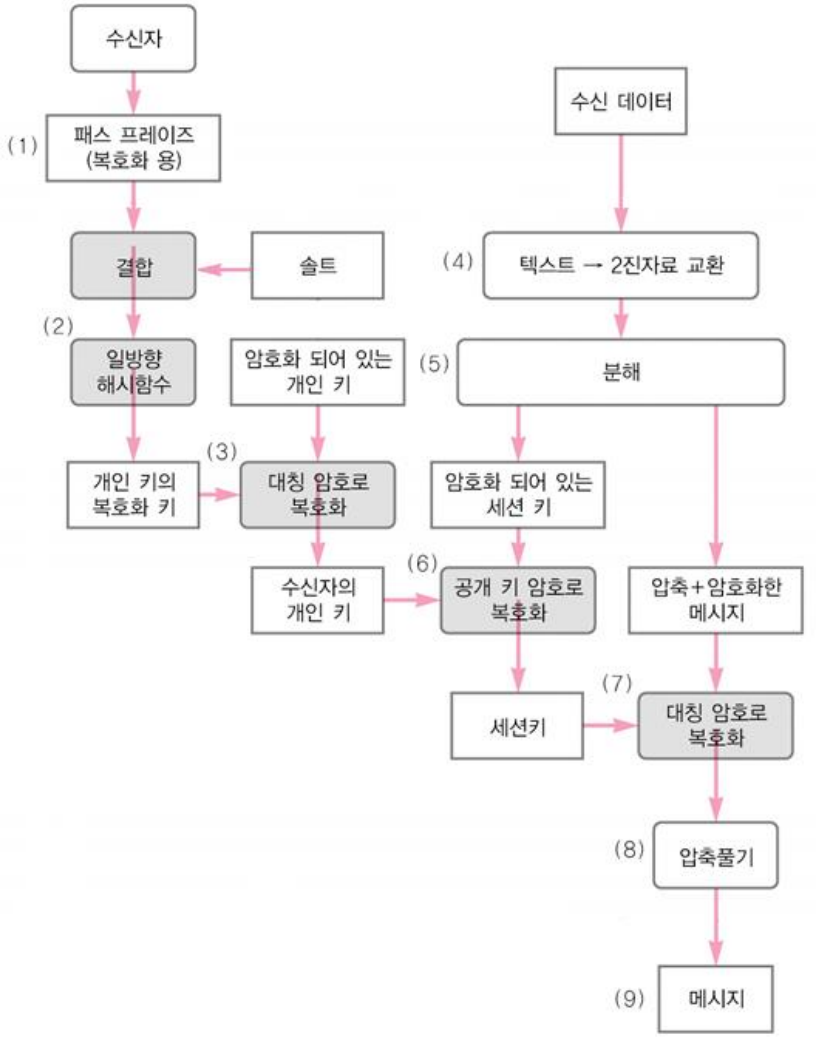
(3) 메시지를 압축

(4) 압축한 메시지를 대칭 암호로 암호화 (절차(1)의 세션 키 사용)

(5) 암호화한 세션 키와, 압축&암호화한 메시지를 결합한다.

(6) 결합된 메시지를 텍스트 데이터로 변환

PGP에 의한 복호화



(1) 수신자는 복호화를 위한 패스 프레이즈를 입력

(2) 패스 프레이즈의 해시 값을 사용하여 개인 키를 복호화 하기 위한 키를 생성

(3) 키 고리 안에 있는 암호화되어 있는 개인 키를 복호화

(4) 수신 데이터(텍스트 데이터)를 이진 데이터로 변환

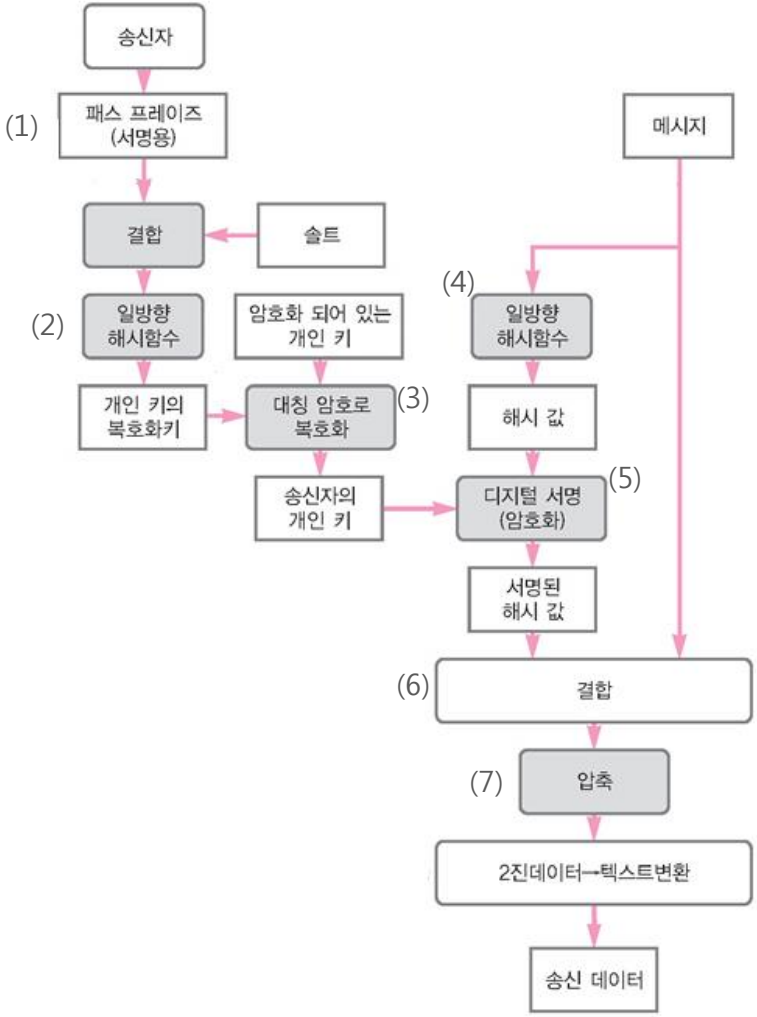
(5) 2진 데이터를 암호화되어 있는 세션 키와 압축 & 암호화 되어있는 메시지로 분해

(6) 암호화되어 있는 세션 키를 공개 키 암호로 복호화 ((3)에서 생성한 수신자의 개인 키 사용)

(7) (5)에서 얻은 압축 & 암호화되어 있는 메시지를 대칭 암호로 복호화 ((6)에서 생성한 세션 키를 사용).

(8) (7)에서 얻은 압축되어 있는 메시지의 압축을 풂

PGP에 의한 디지털 서명 작성



(1) 송신자는 서명을 위한 패스 프레이즈를 입력

(2) 패스 프레이즈의 해시 값을 사용하여 개인 키를 복호화하기 위한 키를 생성

(3) 키 고리 안에 있는 암호화되어 있는 개인 키를 복호화

(4) 일방향 해시 함수를 사용해서 메시지의 해시 값을 계산

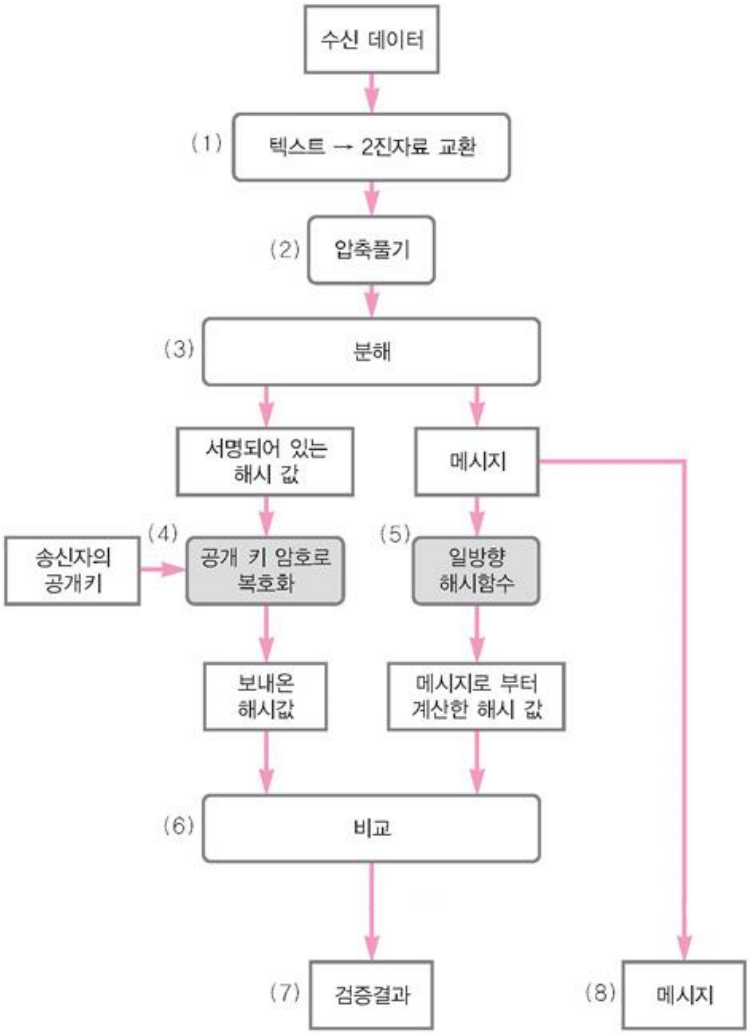
(5) 절차(4)에서 얻은 해시 값에 서명 (송신자의 개인키 사용)

(6) (5)의 작성한 디지털 서명과 메시지를 결합

(7) (6)의 결과를 압축한다.

(8) (7)의 결과를 텍스트 데이터로 변환

PGP에 의한 디지털 서명 검증



(1) 수신 데이터(텍스트 데이터)를 이진 데이터로 변환

(2) 압축되어 있는 데이터의 압축을 풂

(3) 데이터를 서명되어 있는 해시 값과 메시지로 분해

(4) 서명되어 있는 해시 값(암호화되어 있는 해시 값)을 송신자의 공개 키를 사용해서 복호화하고, 보내 온 해시 값을 복원

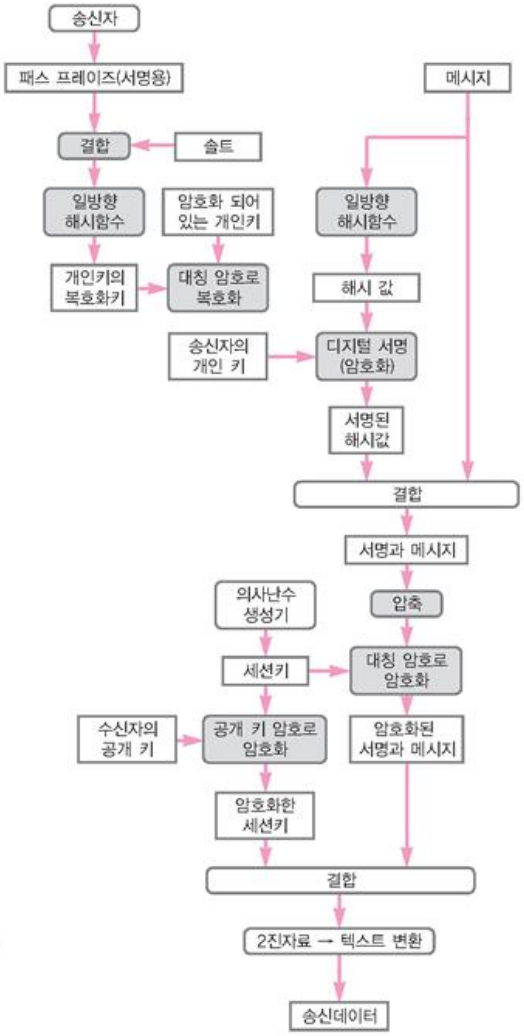
(5) (3)에서 분해한 메시지를 일방향 해시 함수에 부여하여 해시 값을 계산

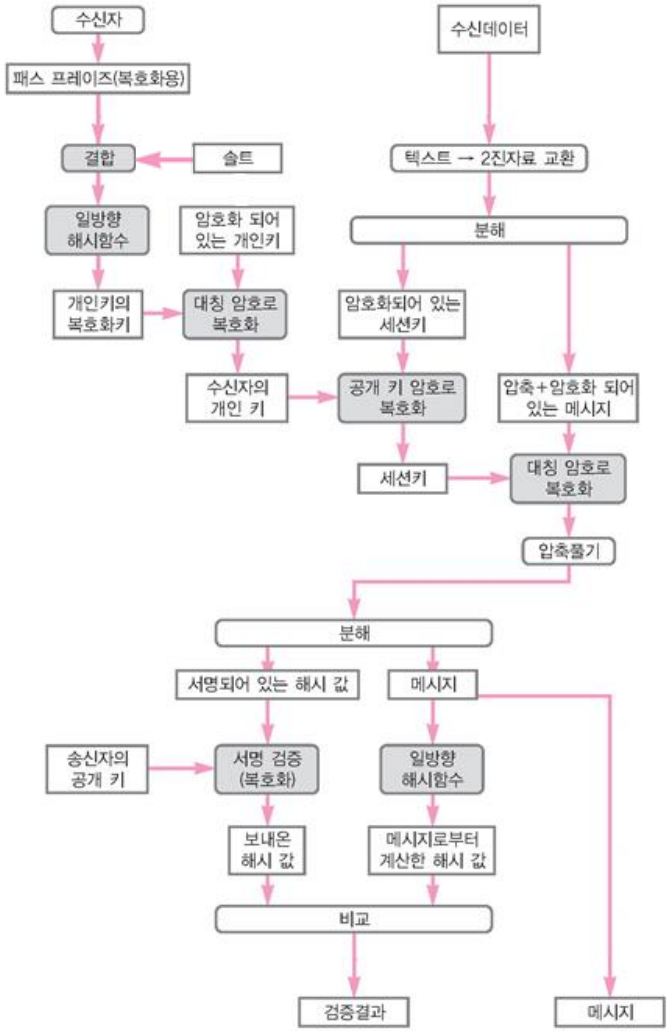
(6) (4)에서 얻은 해시 값과 절차(5)에서 얻은 해시 값을 비교

(7) (6)의 결과가 같으면 디지털 서명의 검증에 성공

(8) (3)에서 분해한 메시지가 송신자의 메시지

PGP 전체 과정


송신자
패스 프레이즈(서명용)
메시지
결합
솔트
일방향 해시함수
암호화 되어 있는 개인키
일방향 해시함수
개인키의 복호화키
대칭 암호로 복호화
해시 값
송신자의 개인 키
디지털 서명 (암호화)
서명된 해시값
결합
서명과 메시지
의사난수 생성기
압축
세션키
대칭 암호로 암호화
수신자의 공개 키
공개 키 암호로 암호화
암호화된 서명과 메시지
암호화한 세션키
결합
2진자료 → 텍스트 변환
송신데이터


수신자
수신데이터
패스 프레이즈(복호화용)
결합
솔트
텍스트 → 2진자료 교환
일방향 해시함수
암호화 되어 있는 개인키
분해
개인키의 복호화키
대칭 암호로 복호화
암호화되어 있는 세션키
수신자의 개인 키
공개 키 암호로 복호화
압축+암호화 되어 있는 메시지
세션키
대칭 암호로 복호화
압축풀기
분해
서명되어 있는 해시 값
메시지
송신자의 공개 키
서명 검증 (복호화)
일방향 해시함수
보내온 해시 값
메시지로부터 계산한 해시 값
비교
검증결과
메시지

PGP 키의 사용

PGP에서 사용되는 암호화키

- 세션키

  - 하나의 메시지에 대하여 각각 세션키 사용

- 공개키

  - 세션키 암호화에 사용

- 개인키

  - 디지털 서명에 사용

- Pass phrase키

  -저장하는 개인키를 암호화하는데 이용

PGP의 키 링

키 링 (Key Ring)

PGP 사용자들은 개인키 링과 공개키 링을 가짐

- 개인키 링
  - ❖ 사용자의 공개키/ 개인키 쌍을 포함
  - ❖ 키 ID를 통하여 식별
  - ❖ Passphrase로 개인키를 암호화하여 관리
- 공개키 링
  - ❖ 사용자에게 알려진 다른 PGP사용자들의 공개키들을 저장하고 관리
  - ❖ 키 ID를 통하여 식별

➢ 키 ID : 해당 키의 최하위 64비트로 구성

PGP의 키 링

키 링 (Key Ring)

PGP 사용자들은 개인키 링과 공개키 링을 가짐

- 개인키 링

  ❖ 사용자의 공개키/ 개인키 쌍을 포함
  ❖ 키 ID를 통하여 식별
  ❖ Passphrase로 개인키를 암호화하여 관리

개인 키 링테이블 형식				
사용자 ID	Key ID	공개키	암호화된 개인키	타임스탬프
kiwoon@naver.com	AB13.....45	AB13.....45..29	32452398...23	260215-16:00
kiwoon@gmail.com	FA24....12	FA24.....12...22	564A4923...23	260213-12:05

PGP의 키 링

키 링 (Key Ring)

PGP 사용자들은 개인키 링과 공개키 링을 가짐

- 공개키 링

  ❖ 사용자에게 알려진 다른 PGP사용자들의 공개키들을 저장하고 관리
  ❖ 키 ID를 통하여 식별

공개키 링 테이블 형식							
사용자 ID	Key ID	공개키	소유자 신뢰 등급	서명	서명 신뢰	키의 적법성	타임스탬프
Alice@…	…	…	…	…	…	…	…
Bob@…	…	…	…	…	…	…	…

PGP 공개키의 정당성

공개키의 정당성

입수한 공개 키가 정말로 자신이 생각하고 있는 인물의 것인지 어떤지를 판단하는 것은 중요

- PGP에서는 인증기관을 사용하지 않음
- PGP에서는 **신뢰 망**(web of trust)이라는 방법을 이용
  - ✓ PGP 사용자가 **서로의 공개 키에 대해 서로 디지털 서명을 하는 방법**

PGP 공개키의 정당성

신뢰망

- 인증기관을 사용하지 않고 개인끼리 신뢰를 확립
- 자신이 어느 키를 신용할지를 결정 가능
  - ✓ PGP 사용자가 **서로의 공개 키에 대해 서로 디지털 서명을 하는 방법**
- 각 사용자가 공개키의 소유자에 대한 소유자 신뢰 값을 설정

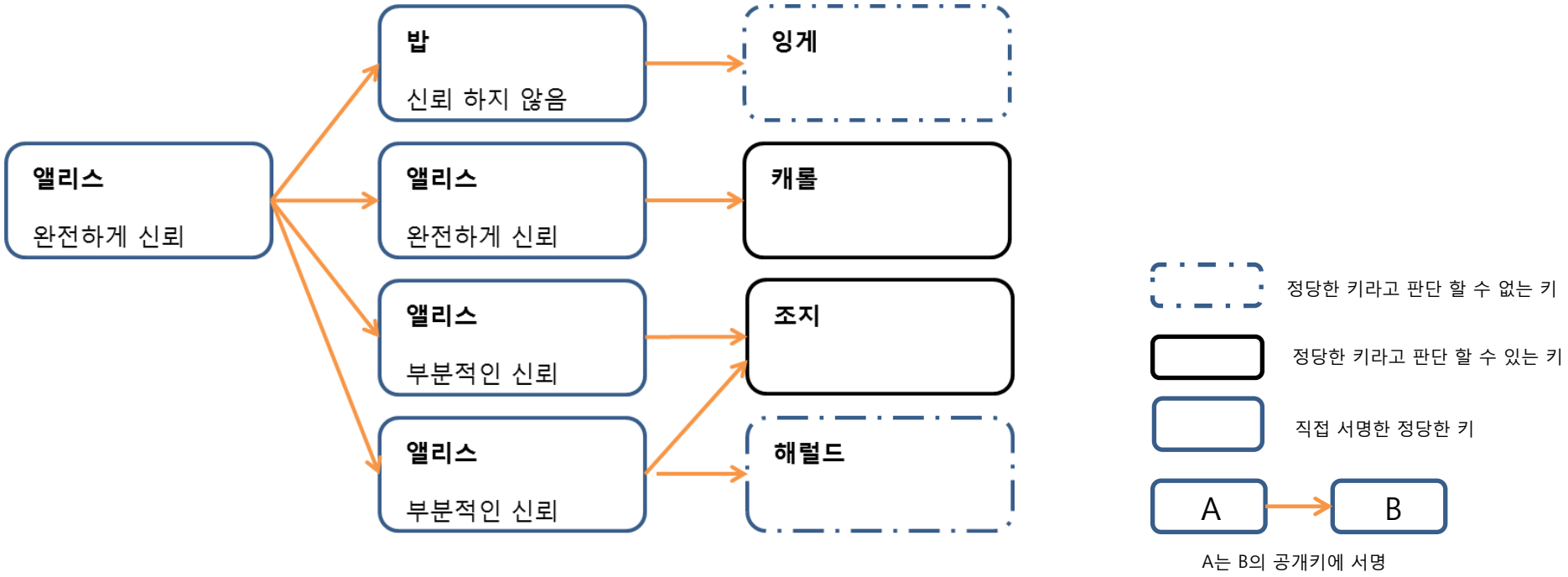
	신뢰 값	중요도
1	신뢰하지 않는 (Nontrusted)	0
2	부분적으로 신뢰하는 (Partial trust)	1/2
3	완전하게 신뢰하는 (Full trust)	1

PGP 신뢰망 구성의 예

케이스1 : 자기 자신의 디지털 서명에 의해 확인

케이스2 : 자신이 항상 신뢰하고 있는 사람의 디지털 서명에 의해 확인

케이스3 : 자신이 부분적으로 신뢰하고 있는 사람들의 디지털 서명에 의해 확인



PGP의 압축

압축 (Compress)

- 메시지 압축을 위해서는 ZIP 알고리즘을 사용
- 서명을 수행한 후에 압축을 취함으로써 암호화된 결과를 가지고 평문을 추측하는 행동을 더욱 어렵게 만듦
- 전자 우편 전송과 파일 저장에 있어 기억 공간을 절약한다는 이점이 있음

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

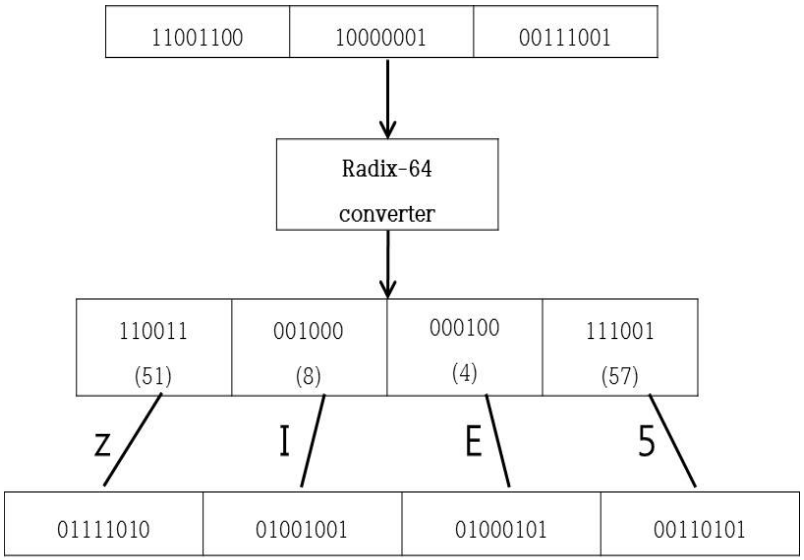
PGP 메일 호환성

메일 호환성 (E-mail compatibility)

- PGP를 사용한 결과로 나오는 메시지 블록의 일부 혹은 전체는 8비트 옥텟 스트림
- 기존의 다른 전자메일 시스템에서는 오직 ASCII 문장으로 이뤄진 블록만 사용해야 하는 제약이 있음
- Radix-64 conversion을 통하여 3개의 8bit를 4개의 ASCII문자로 변환시켜 기존의 전자우편 시스템과의 호환성 문제를 해결

PGP 메일 호환성

Radix-64 변환



value	code	value	code	value	code	value	code	value	code	value	code
0	A	11	L	22	W	33	h	44	s	55	3
1	B	12	M	23	X	34	i	45	t	56	4
2	C	13	N	24	Y	35	j	46	u	57	5
3	D	14	O	25	Z	36	k	47	v	58	6
4	E	15	P	26	a	37	l	48	w	59	7
5	F	16	Q	27	b	38	m	49	x	60	8
6	G	17	R	28	c	39	n	50	y	61	9
7	H	18	S	29	d	40	o	51	z	62	+
8	I	19	T	30	e	41	p	52	0	63	/
9	J	20	U	31	f	42	q	53	1		
10	K	21	V	32	g	43	r	54	2		

PGP의 단편화와 재조립

단편화(Segmentation) & 재조립(Reassembly)

- 전자 우편 프로그램은 대개 50,000byte이하의 메시지를 전송하도록 설정 됨
- PGP에서는 50,000byte이상의 메시지를 단편화하여 전송
- 단편화 과정은 Radix-64 변환을 포함한 다른 과정을 완료한 후 실행
- 분할된 메시지를 재조립

MIME

MIME (Multipurpose Internet Mail Extension)

- SMTP는 실행파일이나 2진 데이터를 전송하지 못함
- 한국어와 같이 2바이트로 구성되는 다중언어, 실행파일, 그림파일과 같은 이진파일도 SMTP로 전송될 수 있도록 ASCII코드로 변환하는 방식
- 이진데이터들을 6비트씩 분할한 후 이를 ASCII 문자로 변환하는 Base64 (또는 Radix 64) 라고하는 코드변환 방식을 사용

MIME

MIME 헤더 필드

- MIME-버전(MIME-Version)
  메시지가 RFC 2045와 2046을 준수(파라메터 값 1.0)

- 내용-타입(Content-Type)
  몸체에 포함된 데이터를 서술

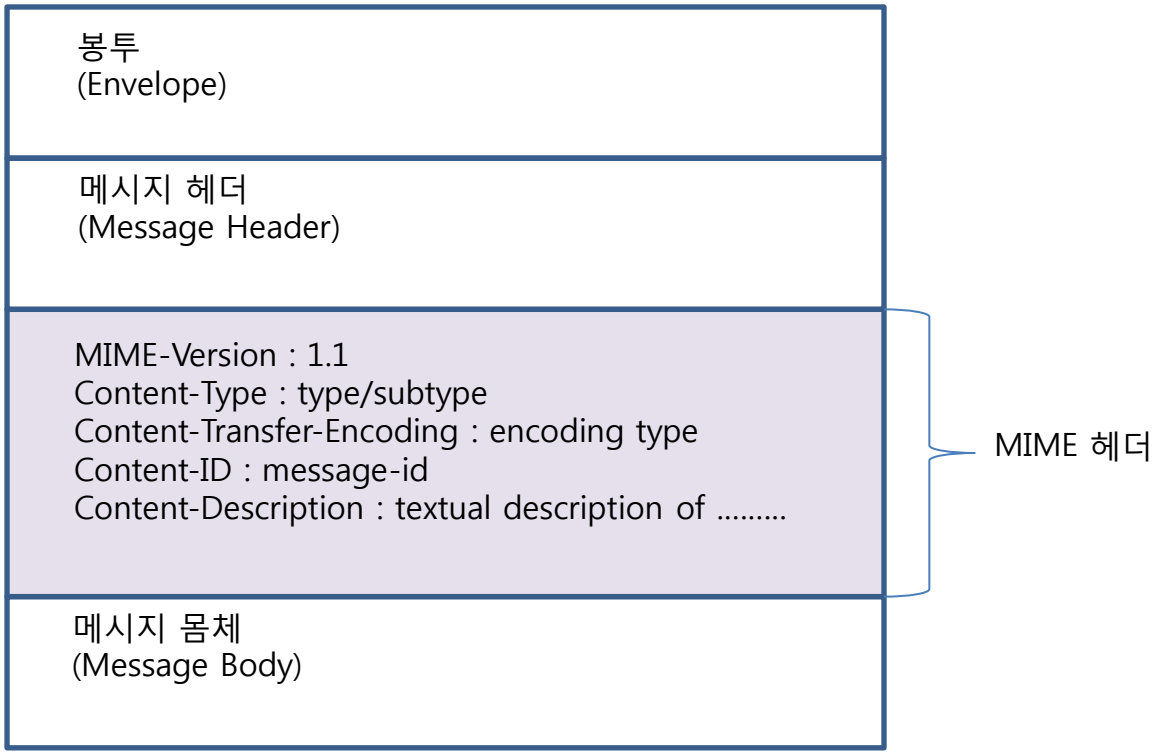
- 내용-전송-부호화(Content-Transfer-Encoding)
  우편전송을 위한 몸체표현의 변형 타입 정의

- 내용-식별자(Content-ID)
  다수의 문맥에서 MIME 엔티티를 유일하게 식별

- 내용-설명(Content-Description)
  몸체를 가진 객체의 텍스트 설명(읽을 수 없는 객체(예:오디오 데이터)에 유용)

MIME

MIME 메시지 포맷



PGP 메일 호환성

MIME 내용 유형

- 7개의 주 타입과 15개의 부 타입으로 구성됨

  주 타입 : 데이터의 일반적인 유형을 선언
  부 타입 : 주 타입 내에서의 특정 형식을 지정

- 주 타입의 의미

  Text : 텍스트
  Multipart : 몸체가 다수의 독립적인 파트들을 포함하고 있을 때 지정
  Message : MIME에서 다수의 중요한 기능(단편화/재조립, 외부 데이터 등)을 제공함
  Image : 이미지
  Video : 비디오
  Audio : 오디오
  Application : 응용

PGP 메일 호환성

MIME 내용 유형

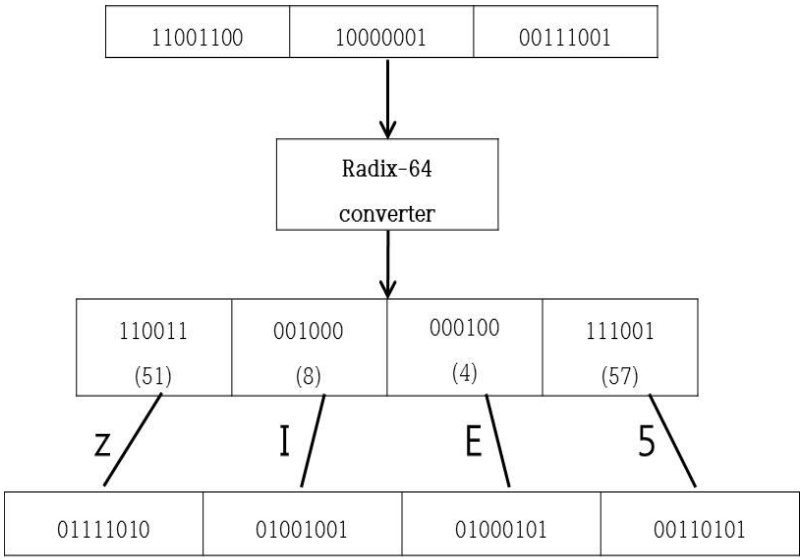
유형	서브유형	설명
텍스트	plain Enriched	포맷되지 않은 텍스트: ASCII나 IOS 8859 다양한 포맷 유연성 제공
멀티파트	Mixed	파트는 서로 독립적이지만 전송은 같이 됨 각 파트는 메일 메시지에 나타나는 순서대로 수신자에게 나타남
	Parallel	Mixed와 비슷하지만 수신자에게 파트를 전달하는 순서가 정의되지 않음
	Alternative	다른 파트는 동일한 정보의 다른버전, 원래 정보에 충실한 정도에따라 순서가 정해지고 수신자의 메일 시스템은 사용자의 가장 좋은 버전을 나타내야만 함
	Digest	Mixed와 비슷하지만 각 파트의 기본 type/subtype은 message/RFC822
메시지	RFC822	Body는 RFC822에 준하는 캡슐화된 메시지
	Partial	수신자에게 큰 메일을 단편화 할 수 있게 함
	External-body	다른 곳에 있는 객체에 대한 포인터를 포함
이미지	jpeg	JPEG 형식 이미지
	gif	GIF 형식 이미지
비디오	mpeg	MPEG 형식 이미지
오디오	basic	표본 추출 비율이 8kHz인 단일-채널 8-비트 ISDN mu-law 부호화
응용	PostScript	Adobe Postscript
	Octet-stream	8-비트 옥텟으로 구성된 일반적인 2진 데이터

PGP 메일 호환성

MIME 메시지 구조의 예

```
MIME-Version: 1.0
    From: Nathaniel Borenstein <nsb@bellcore.com>
    To: Ned Freed <ned@innosoft.com>
    Subject: A multipart example
    Content-Type: multipart/mixed;
        boundary=unique-boundary-1
    This is the preamble area of a multipart message. Mail readers that understand multipart format should ignore this preamble.
      If you are reading this text, you might want to consider changing to a mail reader that understands how to properly display
      multipart messages.

      --unique-boundary-1
        ...Some text appears here...
    [Note that the preceding blank line means no header fields were given and this is text, with charset US ASCII.  It could have
      been done with explicit typing as in the next part.]

    --unique-boundary-1
    Content-type: text/plain; charset=US-ASCII
    This could have been part of the previous part, but illustrates explicit versus implicit typing of body parts.

    --unique-boundary-1
    Content-Type: multipart/parallel;   boundary=unique-boundary-2

    --unique-boundary-2
    Content-Type: audio/basic
    Content-Transfer-Encoding: base64
       ... base64-encoded 8000 Hz single-channel mu-law-format audio data goes here....
```

Radix-64 인코딩



Radix-64 변환표

value	code	value	code	value	code	value	code	value	code	value	code
0	A	11	L	22	W	33	h	44	s	55	3
1	B	12	M	23	X	34	i	45	t	56	4
2	C	13	N	24	Y	35	j	46	u	57	5
3	D	14	O	25	Z	36	k	47	v	58	6
4	E	15	P	26	a	37	l	48	w	59	7
5	F	16	Q	27	b	38	m	49	x	60	8
6	G	17	R	28	c	39	n	50	y	61	9
7	H	18	S	29	d	40	o	51	z	62	+
8	I	19	T	30	e	41	p	52	0	63	/
9	J	20	U	31	f	42	q	53	1		
10	K	21	V	32	g	43	r	54	2		

MIME

S/MIME (Secure/Multipurpose Internet Mail Extension)

MIME에 암호방식을 적용하여 보안성을 제공하는 확장 형태

- S/MIME에서 암호 및 서명에 사용되는 알고리즘

  해쉬 알고리즘: SHA-1, MD5
  서명 알고리즘: DSS, RSA
  세션키 분배방식: Diffie-Hellman, RSA
  대칭키암호(화) 알고리즘: 3DES, RC2/40비트

MIME

S/MIME (Secure/Multipurpose Internet Mail Extension)

MIME에 암호방식을 적용하여 보안성을 제공하는 확장 형태

- S/MIME에서 암호 및 서명에 사용되는 알고리즘

  해쉬 알고리즘: SHA-1, MD5
  서명 알고리즘: DSS, RSA
  세션키 분배방식: Diffie-Hellman, RSA
  대칭키암호(화) 알고리즘: 3DES, RC2/40비트

S/MIME 요구사항 명시

S/MIME은 여러 가지 암호학적 일고리즘을 지정

"must" 는 절대적인 요구사항이며 "should" 는 권고를 의미

메시지 다이제스트 생성	MUST: SHA-1 SHOULD: MD5
메시지 다이제스트 암호화	MUST: DSS SHOULD: RSA SHOULD: 512~1024 비트 키 RSA 서명확인
메시지와 함께 전송을 위한 세션키 암호화	MUST: Diffie-Hellman SHOULD: 512-1024 비트 키 RSA 암호화 지원 SHOULD: RSA의 복호화 지원
일회용 세션키로 전송을 위한 메시지 암호화	SHOULD: 3중 DES와 RC2/40 MUST: 3중 DES 사용 복호화 지원 SHOULD: RC2/40을 갖고 복호화 지원

S/MIME 기능

- **봉함 데이터 (Enveloped Data)**

  임의의 타입 데이터의 암호화된 내용과 하나 이상의 다수의 수신자를 위한 암호화된 내용 암호화 키들로 구성

- **서명된 데이터(Signed Data) :**

  디지털 서명은 서명될 내용의 메시지 다이제스트로부터 만들어져 서명자의 개인키로 암호화 서명과 내용은 base64 방식으로 부호화되며, 서명된 데이터 메시지는 S/MIME 기능을 가진 수신자만 볼 수 있음

- **순수한 서명 데이터(Clear-signed Data)**
  서명 데이터에서와 마찬가지로 내용의 디지털 서명이 만들어지나,
  디지털 서명만을 base64를 이용하여 부호화
  결과적으로 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 봉인된 데이터

envelopedDATA MIME

- ❖대칭 암호 알고리즘을 위한 의사 랜덤 세션키 생성
- ❖RSA 수신자 공개키를 갖고 세션키 암호화
- ❖수신자 정보 RecipientInfo={ 송신자 공개키인증서, 세션키 암호화 알고리즘 식별자, 암호화된 세션키} 블록을 준비
- ❖세션키를 갖고 메시지 암호화

암호화된 메시지의 복호단계

- ❖수신자는 인코딩된 base64를 디코딩
- ❖수신자는 개인키로 암호화된 세션키를 복호화
- ❖메시지 내용을 세션키를 사용하여 복호화

S/MIME 봉인된 데이터

봉함 데이터의 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

S/MIME 서명된 데이터

signedDATA MIME 실체 준비단계

- ❖메시지 다이제스트 알고리즘 선택(SHA, MD5)
- ❖서명할 내용의 메시지 다이제스트 계산
- ❖서명자의 개인키로 메시지 다이제스트를 암호화
- ❖서명자 블록 SignerInfo={서명자의 공개키 인증서, MD 알고리즘 식별자, MD를 암호화하기 위한 알고리즘 식별자, 암호화된 MD}를 준비

서명된 메시지의 검증단계

- ❖수신자는 인코딩된 base64를 디코딩
- ❖서명자의 공개키를 사용하여 MD를 복호
- ❖수신자는 독립적으로 MD를 계산하고 위에서 복호된 MD와 비교

S/MIME 서명된 데이터

서명된 데이터의 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

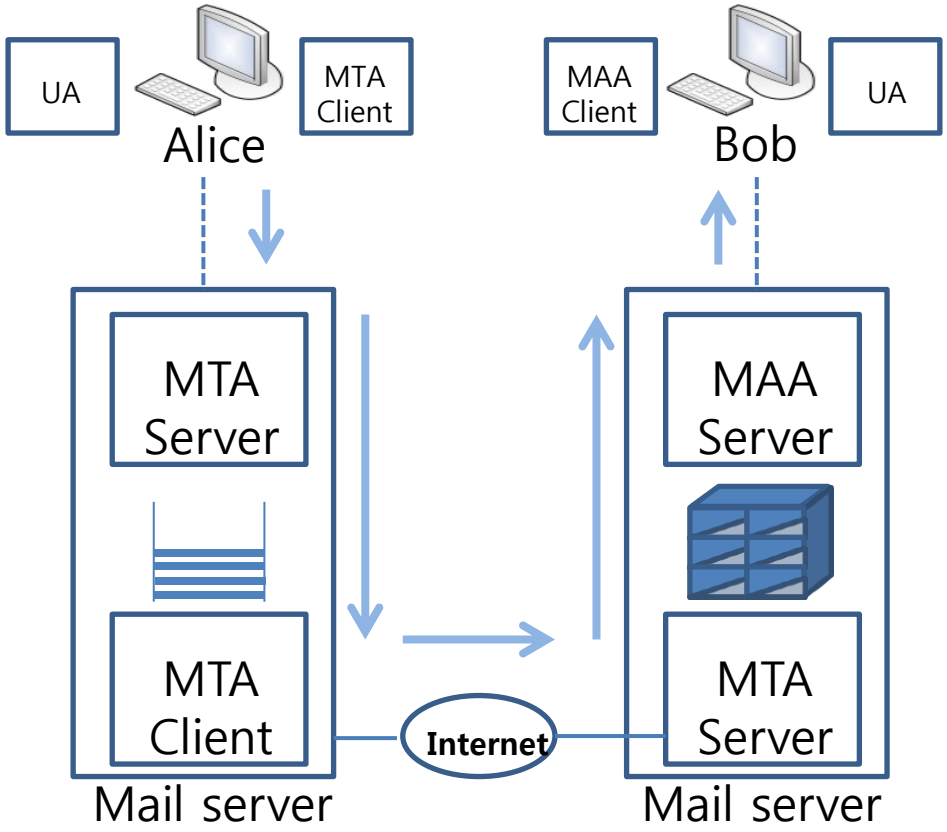
S/MIME 순수한 서명 데이터

순수한 서명 데이터의 예

```
Content-Type: multipart/signed;
   protocol="application/pkcs7-signature";
   micalg=sha1; boundary=boundary42

--boundary42
Content-Type: text/plain

This is a clear-signed message.

--boundary42
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s

ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfHfYT6
4VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbB9HGTrfvbnj
n8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
7GhIGfHfYT64VQbnj756
--boundary42-
```

THANK YOU

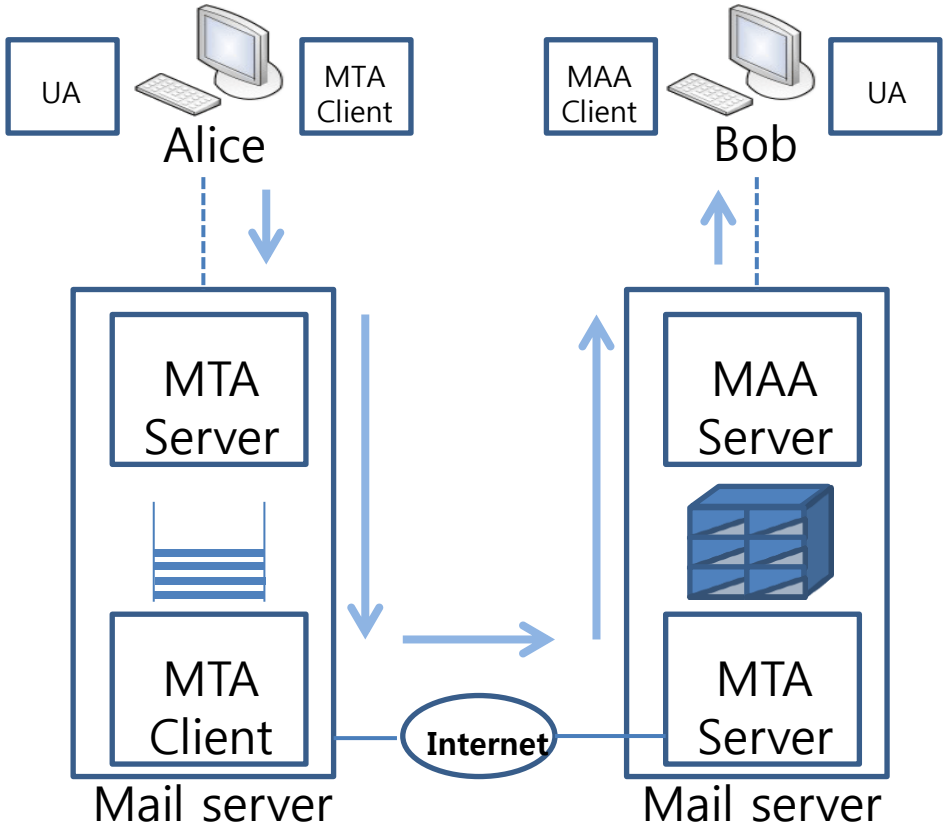
전자우편 구조



메일 전송

- 메일 클라이언트로부터 전달된 메일은 메일 서버의 스풀(Spool)에 저장
- 송신을 위해 스풀에 저장된 메일은 MTA(클라이언트)에 의해 수신자가 연결된 메일 서버의 MTA(서버)로 전송
- 수신자 메일 서버의 고장 등으로 인해 메일을 전송할 수 없으면 해당 메일을 송신자의 메일 서버의 스풀에 일정 시간 동안 저장
- MTA 클라이언트는 약 30분 마다 메일 전송을 재시도
- 정해진 시간 동안 모든 재시도가 실패하면 송신자 메일 서버는 송신자 메일 클라이언트에게 메일 전송 실패를 통보

전자우편 구조



메일 수신

- 송신자 MTA 클라이언트로부터 수신된 메일은 수신자 MTA 서버에 의해 수신자 별로 정해진 메일박스에 저장
- 수신자 별 메일박스의 메일은 수신자 사용자 에이전트에 의해 수신되고 관리
- 수신자 사용자 에이전트는 수신자 메일 서버의 메일박스에 저장된 메일을 편리한 시간에 수신 가능