정보보호와 시스템보안

암호에 대한 이해

전은아

목차

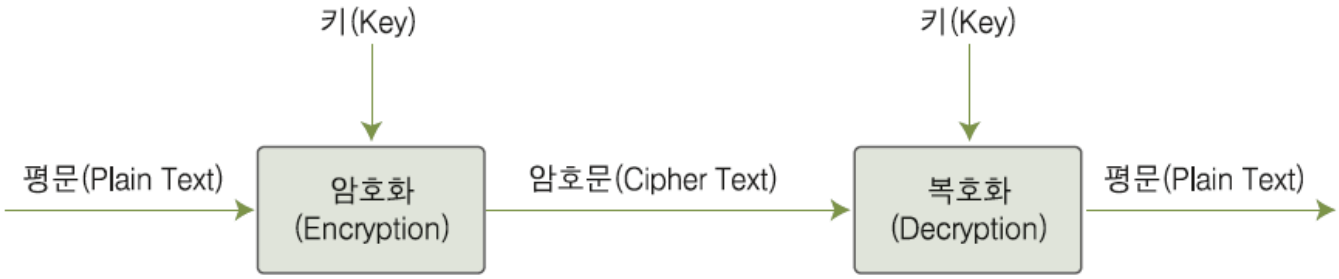
학습목표

- 고전 암호를 통해 암호의 원리를 이해한다.
- 대칭 암호화를 이해한다.
- 비대칭 암호화를 이해한다.
- 비대칭 암호화의 원리와 기능을 이해한다.
- 해시 알고리즘의 원리를 이해한다.

01 암호의 발전사

암호와 관련된 기본 용어

- 암호문(Cipher Text) : 비밀을 유지하기 위해 당사자끼리만 알 수 있도록 꾸민 약속 기호
- 평문(Plain Text) : 암호와 반대되는 말, 누구나 알 수 있게 쓴 일반적인 글
- 암호화(Encryption) :평문을 암호문으로 바꾸는 것
- 복호화(Decryption) : 암호문을 평문으로 바꾸는 것
- 암호화 알고리즘(Encryption Algorithm) : 암호화를 수행하거나, 복호화를 수행할 때 양쪽이 서로 알고 있어야 할 수단
- 암호화키(Encryption Key) : 약속한 규칙



[그림1] 암호화와 복호화

01 암호의 발전사

최초의 암호

- BC 480년 : 스파르타에서 추방되어 페르시아에 살던 데마라토스가 페르시아의 침략 계획 소식을 나무판에 조각하여 적은 후 밀납을 발라 스파르타에 보낸 것
- 스테가노그래피(Steganography) : 실제로 전달하고자 하는 정보 자체를 숨기는 것
  - '덮다'는 뜻의 그리스어'스테가노스(Steganos)'와 '쓰다'라는 뜻의 그라페인 (grapein)이 합쳐진 말

전치법

- 단순히 메시지에 있는 문자의 위치를 바꾸는 방법
- BC 400년 : 스파르타 사람이 군사용으로 사용하던 암호화 방식도 전치법
- 일정 굵기의 봉에 종이를 두르고, 여기에 전달하고자 하는 문장을 쓴 뒤 종이를 풀어 다른 부대에 전달함. (이때 봉의 굵기를 함께 알려줌) 종이를 전달받은 부대는 이를 같은 굵기의 봉에 두른 후에 암호문을 읽음.
  - 종이를 봉에 두르는 것이 암호화 알고리즘, 봉의 굵기가 암호화 키

[그림2] 스파르타의 봉 암호화

01 암호의 발전사

대체법

- 대체법(Substitution)은 해당 글자를 다른 글자로 대체하여 암호화하는 방법
- **단일 치환**
  - 알파벳 한 글자를 다른 하나의 글자로 대체하는 방식으로 암호화를 수행
  - **시저 암호** : BC 50년에 로마 시대의 줄리어스 시저(Julius Caesar)가 군사적인 목적으로 대체법을 사용
    알파벳 26글자를 3자 또는 4자씩 오른쪽으로 이동시킨 뒤 해당되는 글자로 변환시켜 암호화

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w

[그림3] 알파벳을 3자씩 오른쪽으로 이동시킨 결과

  - EHFDUHIXO IRU DVVDVVLQDWRU
    → BE CAREFUL FOR ASSASSINATOR(암살자를 주의하라)

[그림4] 줄리어스 시저

01 암호의 발전사

대체법

단일치환

- **모노 알파베틱 암호** : 알파벳 26자를 각각 다른 알파벳에 대응시키는데, 규칙 없이 임의의 문자에 임의의 알파벳을 대칭시켜 암호화함.
  - 이렇게 만들어진 암호문은 26!(26×25×24...×2×1~4×1026)가지의 경우의 수를 가짐.
  - 간단한 키워드나 키프레이즈(Keyphrase)를 이용해 해당 알고리즘으로 대칭표를 만들기도 함.

- 모노 알파베틱 예 : ASSASSINATOR라는 키워드의 대칭표
  - 키워드에서 중복된 알파벳을 제거하면 ASINTOR. 이 단어를 앞에 놓고(❶), ASINTOR의 마지막 알파벳 R부터 Z까지를 뒤에 적는데 앞에 나온 알파벳은 제외(❷). 다시 A부터 시작해 중복된 알파벳을 제외해 끝까지 적음(❸).

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
A	S	I	N	T	O	R	U	V	W	X	Y	Z	B	C	D	E	F	G	H	J	K	L	M	N	P

❶ ❷ ❸

[그림5] ASSASSINATOR 알파벳 대칭표

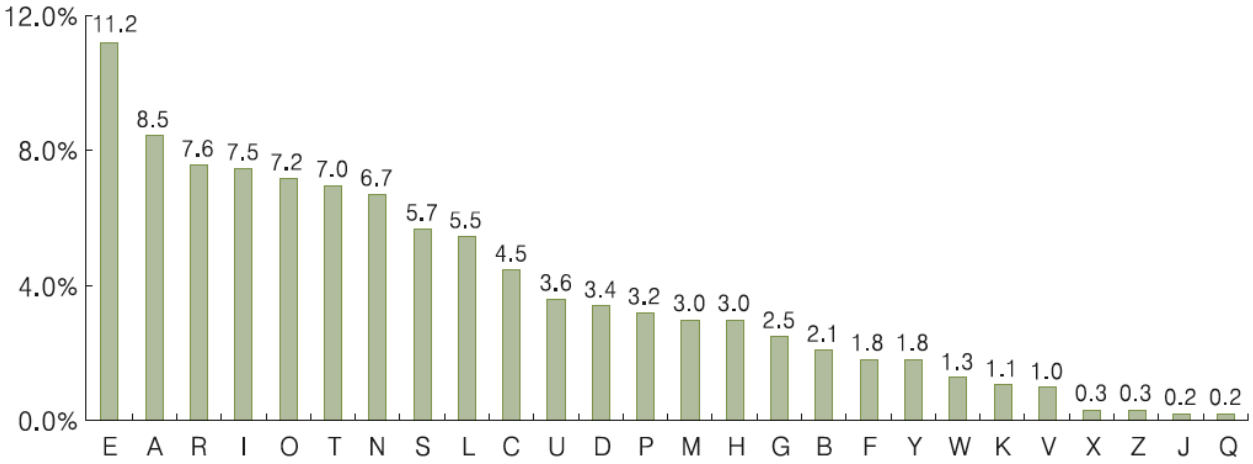
- 단일 치환 암호법은 키워드를 몰라도 복호화가 가능
- 9세기 : 알 킨디라는 아랍의 학자가 기술한 책에 그 복호화 방법이 기록되어 있는데 빈도 분석법(Frequency Analysis)을 이용함.
  - 빈도 분석법은 알파벳의 26자가 문장에 통계적으로 비슷한 빈도 수를 가진다는 점에서 착안한 것

01 암호의 발전사

대체법

단일치환

- **모노 알파베틱 암호**
- 1995년에 출간된 옥스퍼드 영어 사전에서 각 알파벳의 빈도수를 통계낸 것을 살펴보자.



[그림6] 옥스퍼드 영어 사전(9판)의 알파벳별 빈도 수

- 암호문에서 가장 많이 쓰인 알파벳이 T이고, 그 다음이 S, K, G라면 다음과 같은 대칭표가 만들어짐.

E	A	R	I	O	…
T	S	K	G	…	…

[그림7] 알파벳 빈도수별 대칭표

01 암호의 발전사

대체법

다중치환

- 한 글자가 암호화키와의 맵핑에 따라 여러 가지 다른 문자로 대체되어 암호화되는 방식
- **비즈네르 암호화** : 26×26의 알파벳 대칭표를 이용해서 암호화하고자 하는 평문과 암호화키의 맵핑을 이용하여 암호화와 복호화를 수행하는 방식

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
b	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
x	X	Y	z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

[그림9] 비즈네르 표

01 암호의 발전사

대체법

비즈네르 암호화

- 비즈네르 암호화의 예 : 평문은 wish to be free from myself이고, 암호화키는 secret is beautiful

W	i	s	h	t	o	b	e	f	r	e	e	f	r	o	m	m	y	s	e	l	f
s	e	c	r	e	t	i	s	b	e	a	u	t	i	f	u	l	s	e	c	r	e
O	M	U	Y	X	H	J	W	G	V	E	Y	Y	Z	T	G	X	Q	W	G	C	J

[그림10] 비즈네르 암호화 예

- 비즈네르 복호화 과정 : 암호화키의 첫 번째 문자 s를 비즈네르 표의 가로 축으로 하여 O를 찾은 뒤, 세로 축 w를 찾는 방식

s	e	c	r	e	t	i	s	b	e	a	u	t	i	f	u	l	s	e	c	r	e
O	M	U	Y	X	H	J	W	G	V	E	Y	Y	Z	T	G	X	Q	W	G	C	J
w	i	s	h	t	o	b	e	f	r	e	e	f	r	o	m	m	y	s	e	l	f

[그림10] 비즈네르 암호화 예

- 비즈네르 암호화 방식은 17~18세기에 널리 보급되어 사용되었음.
- 19세기에 찰스 배비지가 빈도 분석법을 이용해 규칙성을 찾는 방법으로 복호화 방법을 만듦.

01 암호의 발전사

대체법

다중치환

- **플레이페어 암호**
  - 1854년 찰스 휘트스톤(Charles Wheatstone)이 개발
  - 라이언 플레이페어(Lyon Playfair)를 통해 널리 알려지게 됨.
  - 그의 이름을 따서 플레이페어 암호(Playfair cipher)라고 불림.
  - 1차 세계대전 당시 영국 육군에서 야전 표준 시스템으로 사용
  - 2차 세계대전에는 미 육군 및 기타 연합군에 의해 사용


[그림12] **찰스 휘트스톤(좌)과 라이언 플레이페어(우)**

- 플레이페어 암호화는 2개로 이뤄진 문자 쌍을 다른 문자 쌍으로 대체하는 암호화 방법
  - 보통 정사각형 암호판 안에 영어 알파벳을 배열한 것으로 대체하여 만듦.
  - 암호화키(ASSASSINATOR)에서 중복 문자를 제거한 문자(ASINTOR)를 5×5 정사각형에 순서대로 배열하고, 나머지 알파벳을 차례대로 배열하면 암호판이 완성
  - 이때 5×5 암호판의 칸이 알파벳 개수(26)보다 한칸 모자라므로 I와 J 혹은 Q와 Z를 같은 칸에 넣음.

A	S	I	N	T
O	R	B	C	D
E	F	G	H	J
K	L	M	P	Q/Z
U	V	W	X	Y

[그림13] **플레이페어 암호화 테이블**

01 암호의 발전사

■ 대체법

▪ 플레이페어 암호

- 플레이페어 방식으로 암호화하려면 먼저 주어진 평문을 2개씩 묶은 문자 쌍으로 만들어야 함.
- 띄어쓰기는 무시하고 2개의 문자 쌍을 한 칸씩 차례대로 나열
  - 이때 SS와 같이 한 쌍의 문자가 같거나 마지막에 하나 남은 문자에는 X를 추가하여 문자 쌍으로 만듦.

BE	CA	RE	FU	LF	OR	AS	SA	SX	SI	NA	TO	RX

[그림14] **플레이페어 방식으로 암호문 만들기**

- 평문을 대체함으로써 암호문으로 만들어보자.

❶ 암호화하려는 두 문자가 서로 다른 행과 다른 열에 존재할 경우(BE), 암호 문자는 B와 E의 행과 열이 만나는 곳에 위치한 G(B의 같은 열)와 O(E의 같은 열)이다.

A	S	I	N	T
O	R	B	C	D
E	F	G	H	J
K	L	M	P	Q/Z
U	V	W	X	Y

01 암호의 발전사

■ 대체법

▪ 플레이페어 암호

❷ LF와 같이 두 문자가 같은 열에 있다면 대체되는 암호문은 각각 아래쪽에 있는 문자이다.

– 문자 L은 V, 문자 F는 L에 대체되며 맨 아래쪽 문자일 경우에는 같은 열 맨 위의 문자로 대체됨.

A	S	I	N	T
O	R	B	C	D
E	F	G	H	J
K	L	M	P	Q/Z
U	V	W	X	Y

❸ OR과 같이 두 문자가 같은 행에 있다면 대체되는 암호문은 각각 오른쪽에 있는 문자이다.

– 문자 O는 R, 문자 R은 B에 대체되고 맨 오른쪽 문자일 경우에는 같은 행 맨 왼쪽 문자로 대체됨.

A	S	I	N	T
O	R	B	C	D
E	F	G	H	J
K	L	M	P	Q/Z
U	V	W	X	Y

01 암호의 발전사

■ 대체법

▪ 플레이페어 방식

- ❶, ❷, ❸의 규칙에 따라 각 문자열을 암호화한 결과

BE	CA	RE	FU	LF	OR	AS	SA	SX	SI	NA	TO	RX
GO	NO	FO	VE	VL	RB	SI	IS	VN	IN	TS	DA	VC

[그림15] **플레이페어 방식의 암호화 결과**

- 암호화한 플레이페어 암호화를 복호하하는 방법
  - 암호화할 때 사용한 암호판을 이용하여 ❶, ❷, ❸ 규칙의 반대(위쪽, 왼쪽)의 문자로 대체하면 됨.

02 대칭 암호화 방식

암호학적 강도를 높일 때는 혼돈(Confusion)과 확산(Diffusion)을 이용

- 혼돈 : 암호문의 통계적 성질과 평문의 통계적 성질의 관계를 난해하게 만드는 성질
- 확산 : 각각의 평문 비트와 키 비트가 암호문의 모든 비트에 영향을 주는 성질

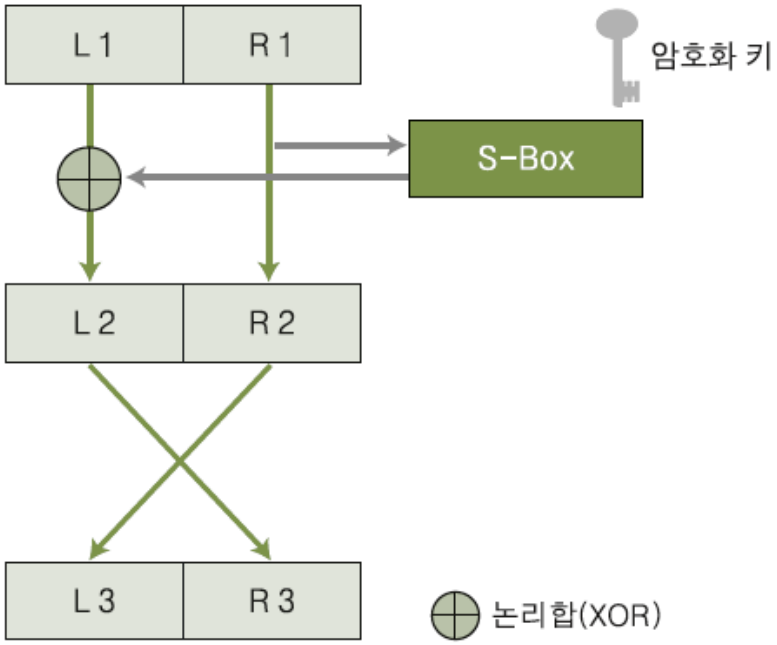
DES (Data Encryption Standard) 알고리즘

- 1972년 미 상무부의 NBS(National Bureau of Standards, 후에 NIST가 된다)에서 보안 문제가 대두됨에 따라 정보보호를 목적으로 공모한 암호 알고리즘.
- IBM의 바터 투흐만(Water Tuchman)과 칼 마이어(Carl Meyer)가 개발
- 1977년 1월 NIST에 의해 암호화 표준으로 결정
- **64비트의 블록 암호화 알고리즘이며, 56비트 크기의 암호화 키로 암호화됨.**
- 생성 가능한 암호화 키는 최대 **256(약 7200조)**가지
- 암호화는 다음 두 가지 기본 변환을 통해 이루어짐.
  ❶ 하나의 블록인 64비트를 L1(32비트)과 R1(32비트)으로 나눔.
  ❷ R1을 암호화 키로 생성한 S-Box로 f 함수를 만들어 치환 작업을 한 후 이 값을 L1과 XOR한 다음 L2와 R2의 위치를 바꿈.

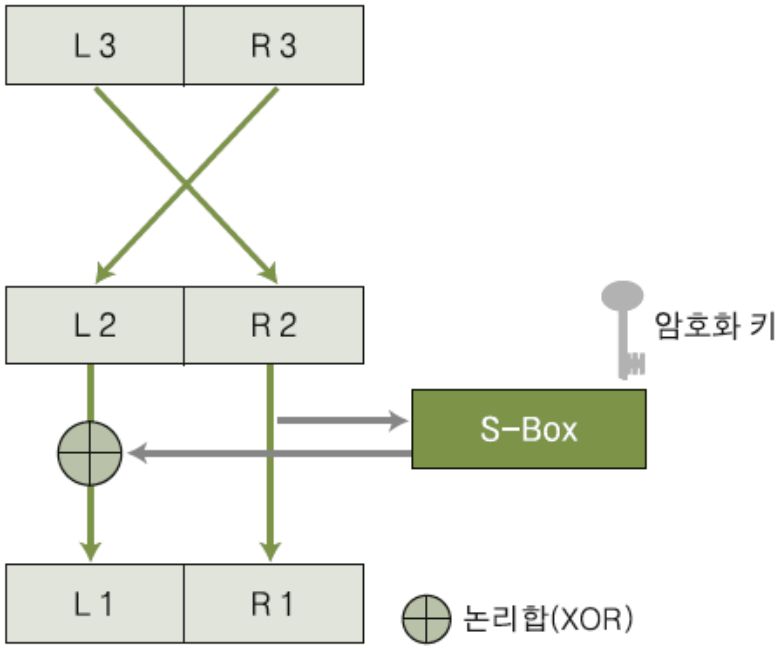
02 대칭 암호화 방식

DES 알고리즘

- 암호화 과정 한 단계를 라운드(Round)라 표현
- 혼돈이 이 과정에서 이루어짐.
- DES는 이러한 과정을 하나의 블록에 대해 알고리즘을 16번 수행하므로 16라운드 알고리즘
- 복호화는 암호화의 반대로 수행



[그림16] DES 암호화 과정

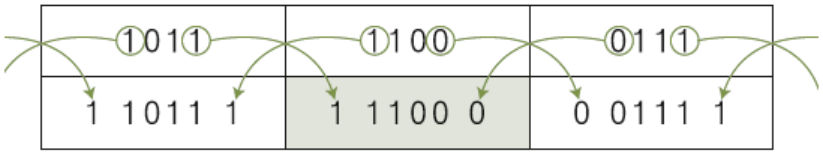


[그림17] DES 복호화 과정

02 대칭 암호화 방식

DES 알고리즘

- S-Box에 넣기 전에 일종의 확장 과정을 거침.



[그림18] DES 암호화 알고리즘의 확장 과정

- S-Box 에서 매칭 111000은 맨 앞 비트 1과 마지막 비트 0을 합쳐 10(2)이 되고, 가운데 블록은 1100(12) 이 됨. 따라서 111000은 위의 S-Box에서 가로 12, 세로 2가 만나는 3(0011)이 됨. 결국 1011 1100 0111의 가운데 있는 1100은 3(0011)으로 암호화됨.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

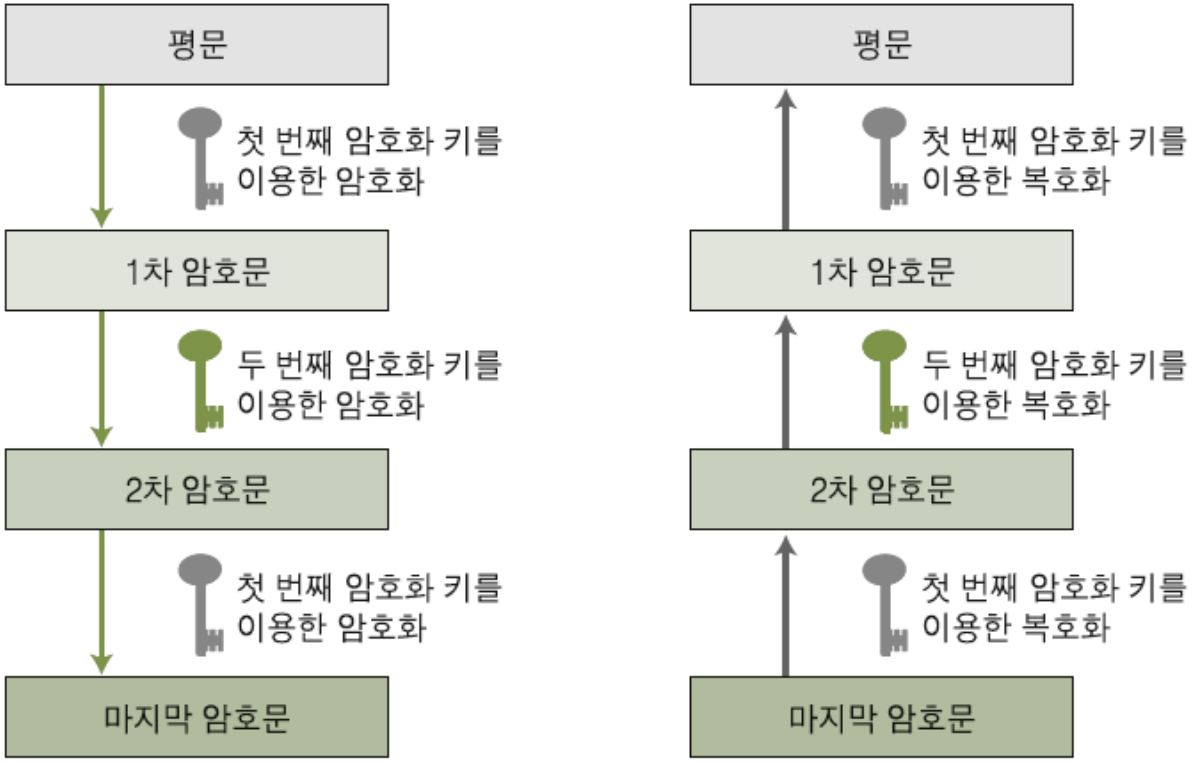
[그림19] DES의 S-BOXㅋ

- DES는 DC(Differential Cryptoanalysis), LC(Linear Cryptoanalysis), DES challenge 등의 공격으로 1999년에 4개월 동안 분산 환경에서 병렬 처리로 복호화하는 데 성공
- 1998년에는 전용 칩을 이용하여 56시간 만에, 1999년에는 전용 칩과 10만 대의 PC를 이용하여 22시간 만에 복호화하는 데 성공
- 1998년 11월 이후부터는 미 정부에서 사용을 중단

02 대칭 암호화 방식

트리플 DES 알고리즘

- **DES의 복호화가 가능해짐에 따라 AES가 나오기 전까지 임시로 사용한 암호화 알고리즘**
- 암호화 및 복호화 과정에서 2개의 암호화키를 이용



[그림20] 트리플 DES 암호화 및 복호화 과정

02 대칭 암호화 방식

AES 알고리즘

- NIST는 1997년 암호화 알고리즘을 다시 공모.
  - 공모 조건은 앞으로 30년 정도 사용할 수 있는 안정성, 128비트 암호화 블록, 다양한 키의 길이.
- 1997년 9월부터 1998년 4월까지 알고리즘 공모를 받았으며 12개국에서 총 15개의 알고리즘이 제안됨.
  - 1998년 8월까지 1차 예선 평가가 이루어져 구현상의 문제점을 검증
  - 1999년 3월까지 효율성 평가를 거쳐 미국의 MARS, RC6, Twofish, 벨기에의 Rijndael, 영국/이스라엘/덴마크의 합작인 Serpent가 결선 알고리즘으로 선정.
- 결선에서는 공개적으로 암호학적 안전성 분석을 하였는데 리즈멘(Rijmen)과 대먼(Daemen)의 Rijndael 알고리즘이 2000년 10월 최종 AES(Advanced Encryption Standard)로 선정.

SEED 알고리즘

- 전자상거래, 금융, 무선통신 등에서 전송되는 개인정보와 같은 중요한 정보를 보호하기 위해, 1999년 2월 한국인터넷진흥원과 국내 암호전문가들이 순수 국내기술로 개발한 128비트 블록의 암호 알고리즘

ARIA 알고리즘

- 전자정부 구현 등으로 다양한 환경에 적합한 암호화 알고리즘이 필요함에 따라 국가보안기술 연구소(NSRI) 주도로 학계, 국가정보원 등의 암호전문가들이 힘을 모아 개발한 국가 암호화 알고리즘
- ARIA 알고리즘은 경량 환경 및 하드웨어에서의 효율성 향상을 위해 개발된 128비트 블록 암호 알고리즘
- 2004년에 국가표준기본법에 의거하고 지식경제부에 의해 국가표준(KS)으로 지정

02 대칭 암호화 방식

기타 대칭형 알고리즘

IDEA

- 1990년 : ETH(Eidgenossische Technische Hochschule)의 라이(Lai)와 매시(Massey)가 제안한 PES(Proposed Encryption Standard)가 발표됨.
- 1991년 : 이를 개선해 IPES(Improved PES)라는 이름으로 다시 발표됨.
- 1992년 : IDEA(International Data Encryption Standard)로 이름이 바뀜.
- IDEA는 128비트의 키를 사용해 64비트의 평문을 8라운드를 거쳐 64비트의 암호문을 생성
- 모든 연산이 16비트 단위로 이루어지도록 하여 16비트 프로세서에서 구현이 용이
- 주로 키 교환에 쓰임.

RC5

- 1994년 미국 RSA 연구소의 리베스트(Rivest)가 개발한 입출력, 키, 라운드 수가 가변인 블록 알고리즘
- RC5(Ron's Code 5)는 32/64/128비트의 키를 가짐.
- 속도는 DES의 10배

Skipjack

- 미 국가안보국(NSA)에서 개발한 Clipper 칩에 내장되는 블록 알고리즘
- 알고리즘의 형태와 구조를 비밀로 유지하다가 1998년에 공개됨
- 소프트웨어로 구현되는 것을 막고자 Fortezza Card에 칩 형태로 구현됨.
- 전화기와 같이 음성을 암호화하는 데 주로 사용
- 64비트의 입출력, 80비트의 키, 총 32라운드를 가짐.

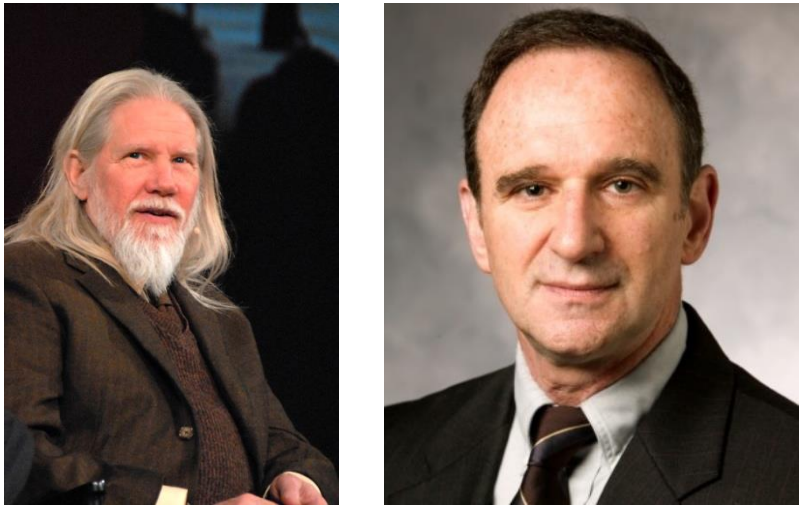
03 비대칭 암호화 방식

등장 배경

- 대칭 암호화 방식으로는 암호화 키 교환의 문제를 해결할 수 없었음.
- 이를 위해 비대칭 암호화 방식이 연구됨.

비대칭 암호화 방식의 발견

- 1974년부터 암호 전달 문제를 연구하기 시작
- 1975년 디피는 비대칭키라 부르는 개념을 집에서 콜라를 가지러 아래층으로 내려가던 중에 떠올림.

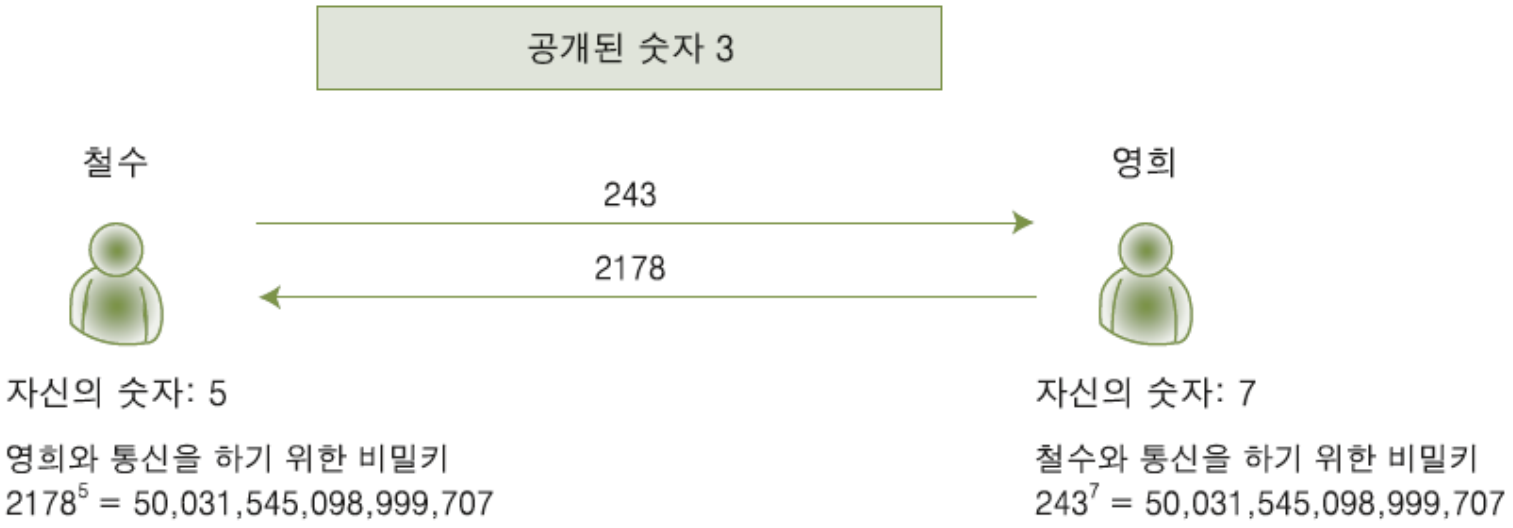


[그림21] 위트필드 디피(좌)와 마틴 헬만(우)

03 비대칭 암호화 방식

비대칭 암호화 방식의 발견

- 공개된 정보가 3이라 가정
- 같은 키를 공유하기 위해 철수는 자신이 정한 숫자 5를 사용해 3^5인 243이라는 수를 영희에게 보냄.
- 영희도 자신의 숫자를 7로 정하고, 3^7인 2,178을 철수에게 보냄.
- 철수와 영희는 상대에게 받은 수에 자신의 수를 제곱승.
- 둘은 자신이 정한 5와 7 숫자를 상대방에게 전달하지 않고서도 50,031,545,098,999,707 이라는 같은 키를 공유하게 됨.

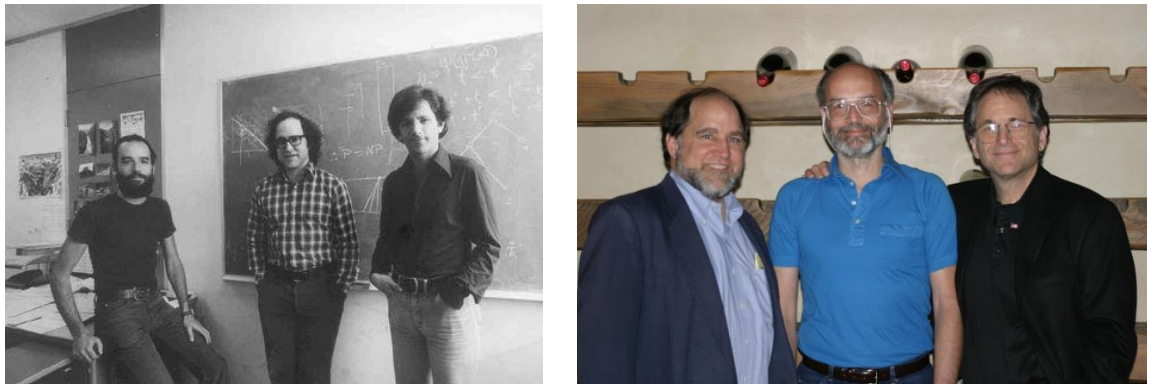


[그림22] 키 공유에 관한 기본 아이디어

03 비대칭 암호화 방식

RSA 알고리즘

- MIT의 로널드 리베스트(Ronald Rivest), 아디 샤미르(Adi Shamir), 레오나르도 애들먼(Leonard Adleman)이 고안



[그림23] 리베스토, 샤미르, 애들먼(과거 모습(좌) 현재 모습(우))

- RSA 암호는 소수(素數)를 이용
  - RSA 암호의 아이디어는 중요 정보를 두 개의 소수로 표현한 후, 두 소수의 곱을 힌트와 함께 전송해 암호로 사용하는 것
  - RSA 알고리즘에서는 모든 사람이 고유한N값을 갖게 됨. (N은 두 소수의 곱)
  - 만약 영희가 자신의 N을 p=17,159와 q=10,247의 곱인 N=17,159*10,247=175,828,273으로 정함.
  - 영희가 자신의 N값을 모든 사람들에게 공개하면 이 때의 N 값은 영희의 공개키가 됨.
  - 영희에게 메시지를 보내고 싶은 사람은 N 값을 찾아 어떤 알고리즘을 통해 암호화를 한 후 영희에게 보냄.
  - 여기에서 p와 q는 영희의 사설키

03 비대칭 암호화 방식

RSA 알고리즘

- 리베스트, 샤미르, 애들먼은 1977년 8월에 미국의 과학잡지인 사이언티픽 아메리칸(Scientific American)에 129자리인 N의 소인수 p와 q를 찾아보라는 퀴즈를 냄.

  N=114,381,625,757,888,867,669,235,779,976,146,612,010,218,296,721,242,362,562,561,842,935,706,935,245,733,897,830,597,123,563,958,705,058,989,075,147,599,290,026,879,543,541

- 잡지에 실린 지 17년 만인 1994년 4월 26일에 600명의 지원자로 이루어진 팀이 p와 q 값을 발견

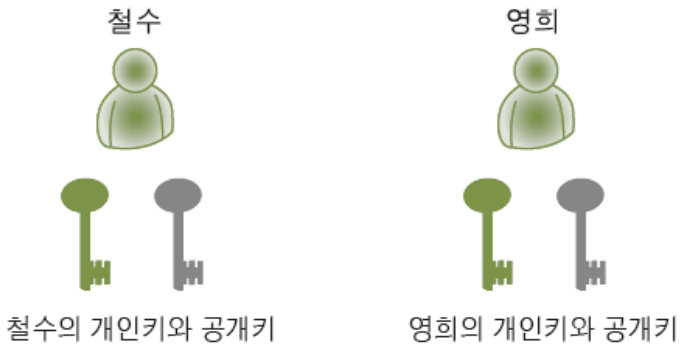
  p=3,490,529,510,847,650,949,147,849,619,903,898,133,417,764,638,493,387,843,990,820,577
  q=32,769,132,993,266,709,549,961,988,190,834,461,413,177,642,967,992,942,539,798,288,533

- 현재 사용되는 250자리 RSA 암호는 복호화하는 데 우주의 나이만큼 소요됨.

03 비대칭 암호화 방식

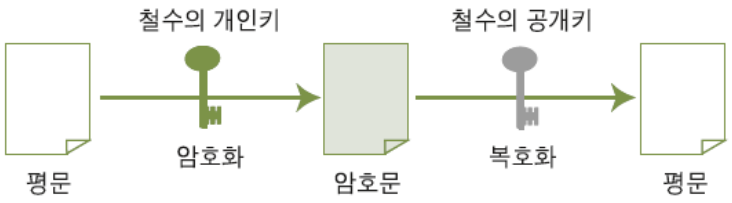
비대칭 암호화의 구조

- 각 개인이 공개키(Public Key)와 개인키(Private Key)를 소유하는 구조

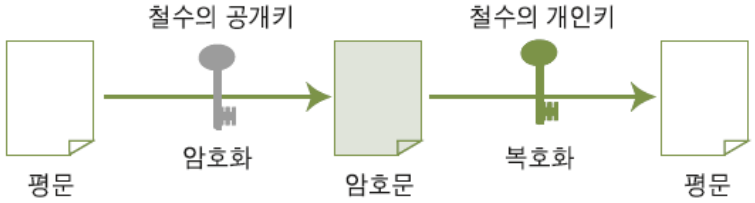


[그림24] 각자 소유하고 있는 공개키와 개인키

- 비대칭 암호화 알고리즘에서는 언제나 한 쌍의 개인키와 공개키에 의해 암호화와 복호화가 이루어짐.
  - 철수의 개인키로 암호화된 메시지는 철수의 개인키로 복호화되지 않고, 오직 철수의 공개키로 복호화됨.
  - 반대로 철수의 공개키로 암호화를 먼저 수행할 수도 있으며, 이런 경우 복호화는 철수의 개인키로만 가능



[그림25] 개인키와 공개키의 관계 1



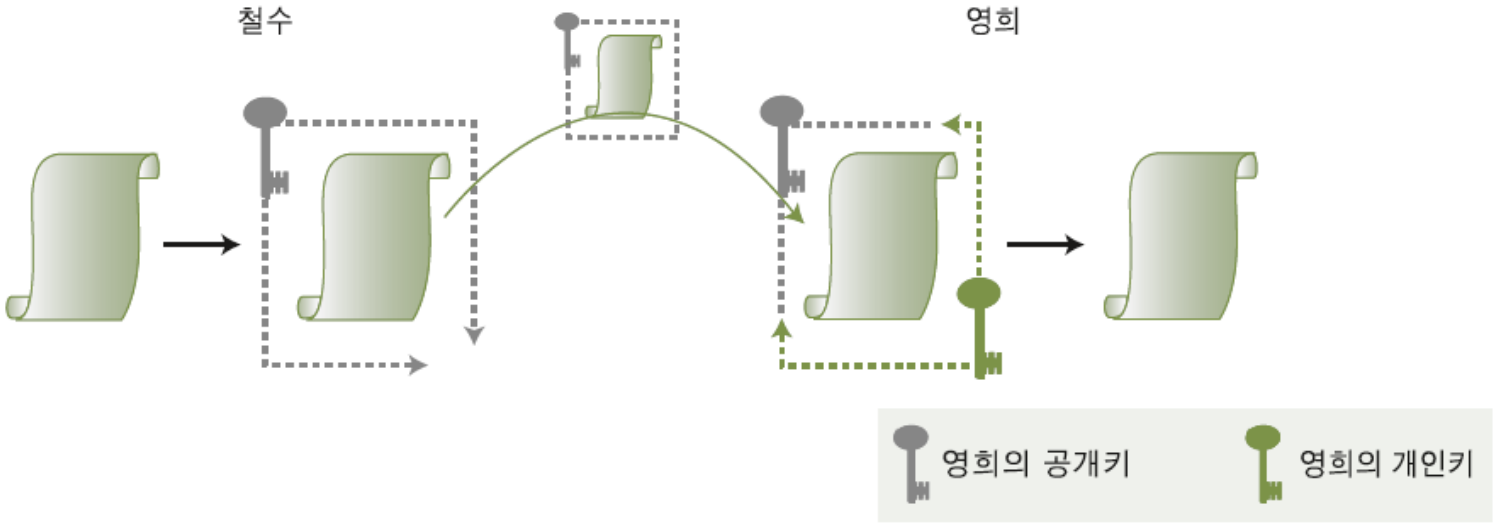
[그림26] 개인키와 공개키의 관계 2

03 비대칭 암호화 방식

비대칭 암호화의 기능

기밀성

- 비대칭 암호화 알고리즘의 가장 기본적인 기능은 기밀성(Confidentiality)
  - 철수는 전화번호부에서 전화번호를 찾듯이 영희의 공개키(Public Key)를 구함
  - 이 공개키를 이용해 편지를 암호화해서 보내면 영희는 자신이 가진 사설키(Private Key)를 이용해 철수의 편지를 복호화하여 읽을 수 있음.



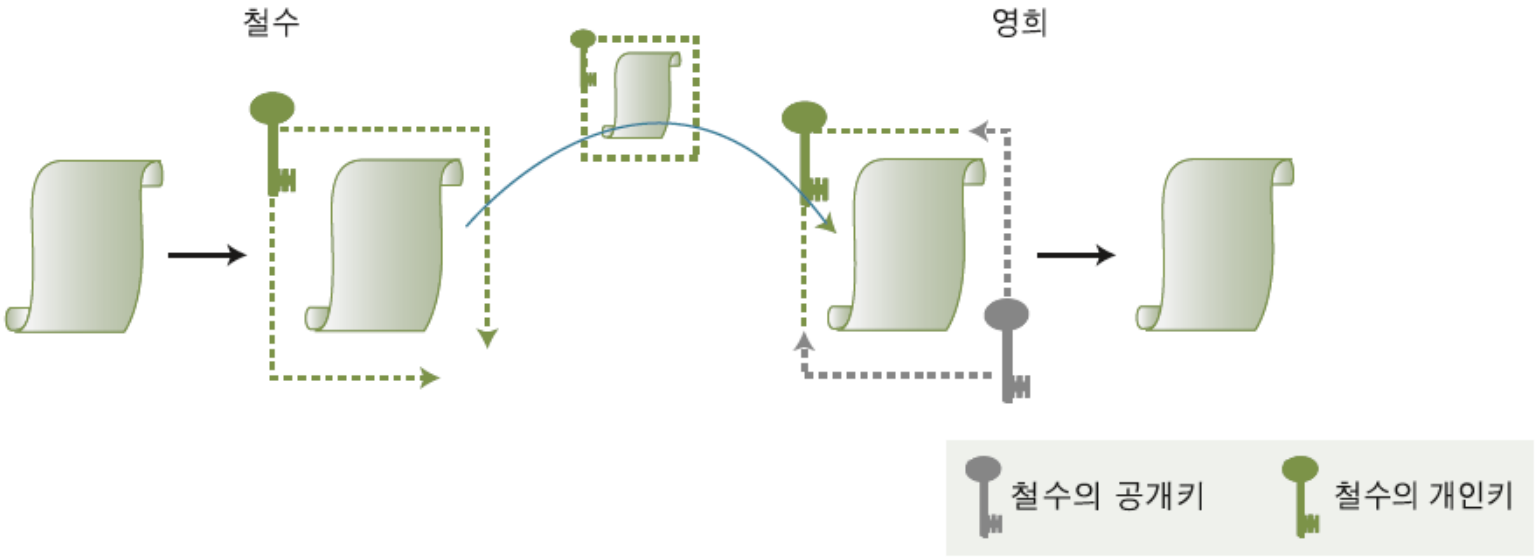
[그림27] 기밀성 확보를 위해 공개키를 이용해 암호화하기

03 비대칭 암호화 방식

비대칭 암호화의 기능

부인 방지

- 철수는 영희에게 편지를 보낼 때 자신의 개인키로 편지를 암호화하여 전송.
- 철수의 개인키로 암호화된 편지는 철수의 공개키로만 열 수 있으므로 영희는 그 편지가 철수가 쓴 것임을 확신할 수 있음.



[그림28] 부인 방지 기능 확보를 위해 개인키를 이용해 암호화하기

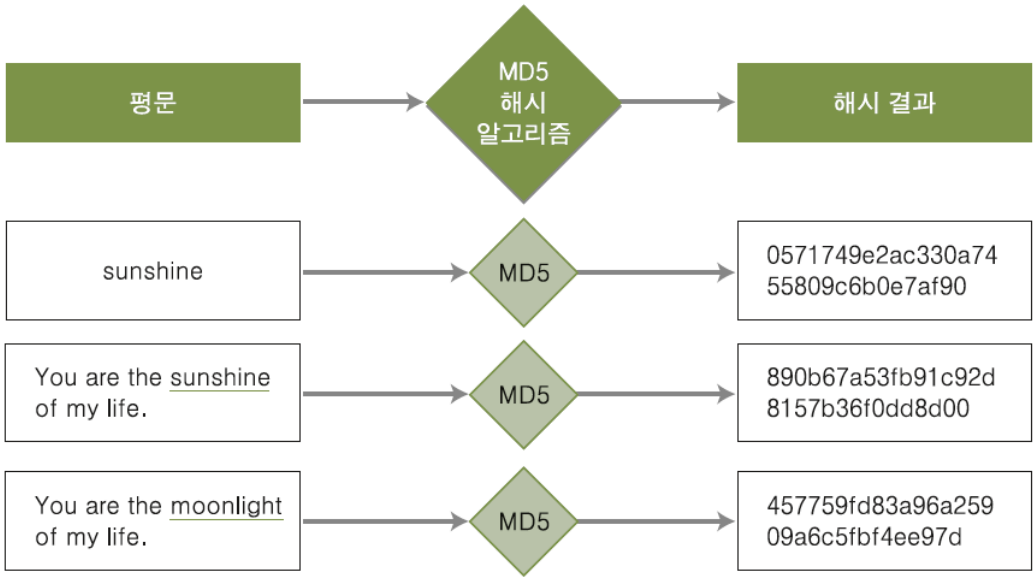
04 해시

해시의 정의

- 하나의 문자열을, 이를 상징하는 더 짧은 길이의 값이나 키로 변환하는 것
- 정보의 위변조를 확인하기 위한 방법 : 무결성

해시의 특징

- 세 평문은 길이가 다르지만 해시 결과는 32개의 문자로 길이가 모두 같음.
- 또한 둘째와 세째 평문은 단어 하나만 다를 뿐인데 해시 결과가 완전히 다름.
- 이와 같은 결과는 해시값을 통해 해시되기 전의 값을 추측하는 것이 불가능하게 하는 해시의 특성 때문임.



[그림29] 각 평문에 대한 MD5 해시값

04 해시

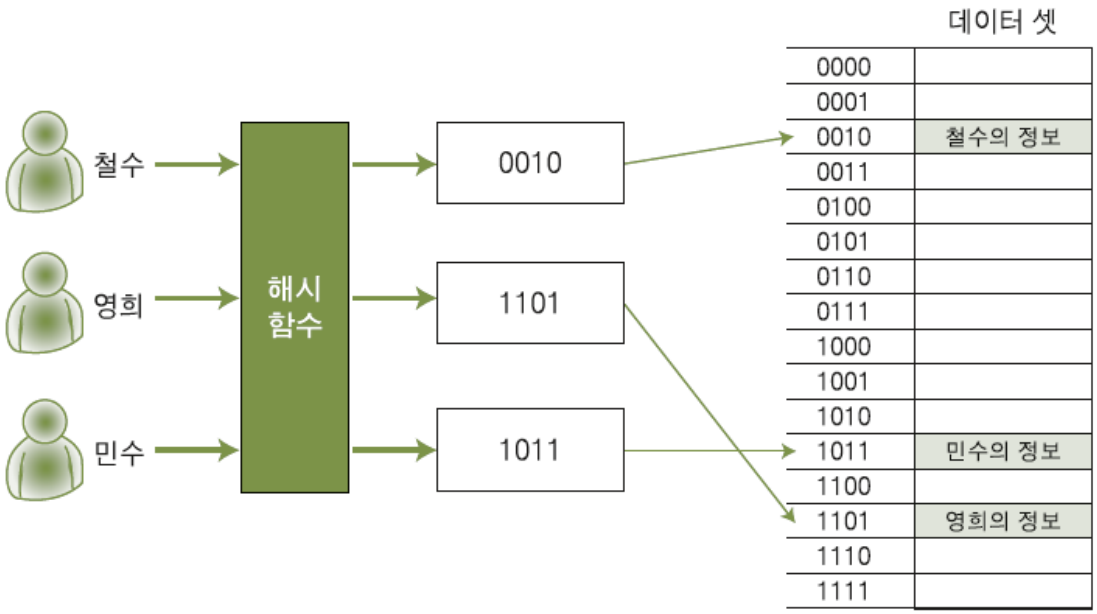
해시의 특징

- MD5
  - 32개의 16진수로 이루어졌음.
  - 16^{32} = 340,282,366,920,938,463,463,374,607,431,768,211,456 개의 결과값이 존재
  - 이 수는 충분히 커 보이지만 무한은 아님.
  - 따라서 다른 데이터를 입력해도 해시 결과값이 같을 수 있음.
    - 이를 충돌(Collision)이라 함.
  - 충돌이 자주 일어나는 해시는 좋은 해시가 아님.

04 해시

해시의 역할

- 해시를 통해 0010이라는 라벨 값을 부여받아 해당 철수의 데이터로 직접 접근이 가능.
- 이렇게 구현된 데이터베이스 탐색 로직은 모든 참조값에 대해 데이터 반환
- 시간이 균일하고 순차 탐색보다 속도가 훨씬 빠름.



[그림30] 데이터베이스에서의 해시값을 통한 값의 참조

- 보안에서는 해시를 무결성 확인을 위한 알고리즘으로 사용
  - 무결성 :오직 허가된 사람들에게만 정보가 개방되고, 그들에 의해서만 수정될 수 있음을 보장한다는 의미

04 해시

■ 해시의 종류

▪ MD 알고리즘

- MD(Message Digest function 95) 알고리즘에는 MD2, MD4, MD5 이렇게 세 가지가 있음.
- RSA를 개발한 미국 MIT의 로널드 리베스트 교수가 공개키 기반 구조를 만들기 위해 RSA와 함께 개발
- 1989년에 만들어진 MD2는 8비트 컴퓨터에 최적화되어 있고, MD4(1990년 개발)와 MD5(1991년 개발)는 32비트 컴퓨터에 최적화되어 있음.
- MD5 알고리즘은 MD4의 확장판으로, MD4보다 속도가 빠르지는 않지만 데이터 보안성에 있어 더 많은 확신을 제공

▪ SHA 알고리즘

- SHA(Secure Hash Algorithm) 알고리즘은 미국 NSA에 의해 만들어짐.
- 160비트의 값을 생성하는 해시 함수로, MD4가 발전한 형태
- MD5보다 조금 느리지만 좀더 안전한 것으로 알려져 있음.
- SHA에 입력하는 데이터는 512비트 크기의 블록
- SHA 알고리즘은 크게 SHA-1과 SHA-2로 나눌 수 있음(SHA-256, 384, 512는 SHA-2에 속한다).

[표1] SHA 알고리즘의 종류와 특징

알고리즘	메시지 문자 크기	블록 크기	해시 결과값 길이	해시 강도
SHA-1	< 2^{46}	512비트	160비트	0.625
SHA-256	< 2^{46}	512비트	256비트	1
SHA-384	< 2^{12}	1024비트	384비트	1.5
SHA-512	< 2^{12}	1024비트	512비트	2

정보보호와 시스템보안

전자상거래보안

전은아

목차

학습목표

- 전자상거래의 보안 요구 사항을 이해한다.
- 공개키 기반기술의 원리를 이해한다.
- 공인인증서에 대해 알아본다.
- 전자상거래에서 이용되는 암호화와 해시 기술을 이해한다.
- PGP를 이용하여 암호화된 메일을 살펴본다.
- IPSEC와 SSL에 대해 알아본다.

01 전자상거래에 대한 이해

전자상거래의 시작

- 1979년 : 마이클 알드리치(Michael Aldrich)는 전화선을 이용해 통신하도록 개조된 TV로 최초의 온라인 쇼핑을 가능하게 함.
- 비디오텍스는 1970년대부터 1980년 중반까지 주로 온라인 홈뱅킹에 사용됨.
- 1981년 : 비디오텍스를 이용하여 씨티뱅크(Citibank), 체이스 맨하탄(Chase Manhattan), 케미컬(Chemical), 메뉴팩쳐러스 하노버(Manufacturers Hanover)가 뉴욕에서 서비스를 제공하기 시작
- 1994년 : 본격적인 전자상거래의 시작
  - 처음으로 피자헛(Pizza Hut)이 웹 페이지를 통해 주문을 받음.
  - 최초의 인터넷을 통한 온라인 서비스를 제공하는 은행(스탠다드 연방 신용 연합, Stanford Federal Credit Union)이 문을 열었으며 코드 네임 모질라(Mozilla)로 넷스케이프(Netscape) 1.0이 만들어짐.
  - 넷스케이프는 SSL 암호화를 통해 안전한 거래를 제공했다.
- 1995년 : 월 스트리트의 컴퓨터 시스템 전문가였던 제프 베조스(Jeff Bezos)가 인터넷 가상 상점인 아마존(Amazon)을 설립

[그림 1] 비디오텍스

[그림 3] 제프 베조스

[그림 4] 1994년 아마존 사이트

01 전자상거래에 대한 이해

■ 전자상거래의 보안 요건

[표 1] 전자상거래의 보안 공격 유형

공격 유형	설명
인증에 대한 공격	네트워크를 통해 접근한 사용자가 적절하지 못한 인증을 통해 다른 사용자로 위장하는 것 예) 최근 가짜 은행 사이트를 만들어 은행 사용자에 대한 공인인증서 정보를 획득하여 악용하는 사례 등
송·수신 부인 공격	네트워크를 통해 수행한 인증 및 거래 내역에 대해 부인하는 것 예) 계좌이체 및 신용카드 지불을 받고도 받지 않았다고 부인하거나, 소매점으로부터 상품을 받은 후 받지 않았다고 부인하는 사례 등
기밀성에 대한 공격	네트워크를 통해 전달되는 인증 정보 및 주요 거래 정보가 유출되는 것 예) 전자 결제 시 카드 번호 정보가 유출되어 부정 사용되는 사례 등
무결성에 대한 공격	네트워크의 도중에 거래 정보 등이 변조되는 것 예) 온라인 계좌이체 등을 이용한 전자 결제 시 수신 계좌나 금액 등을 변조하는 사례 등

- 전자상거래가 성공하기 위한 조건
  - 전자상거래에서는 원격의 거래 상대를 신뢰하기 어려우므로 네트워크상에서 상대방 및 자신에 대한 신분 확인 수단이 필요
  - 전자상거래에서는 거래 사실(거래 내역)의 공증을 보장할 수 있는 신뢰할 만한 제3자의 중재가 필요
  - 전자상거래에서는 전자지불 방식(과정)의 안정성을 보장하기 위한 방법이 확보되어야 함.

02 공개키 기반 구조

공개키 기반 구조의 개념

- 공개키 기반 구조(PKI, Public Key Infrastructure)는 메시지의 암호화 및 전자서명을 제공하는 복합적인 보안 시스템 환경

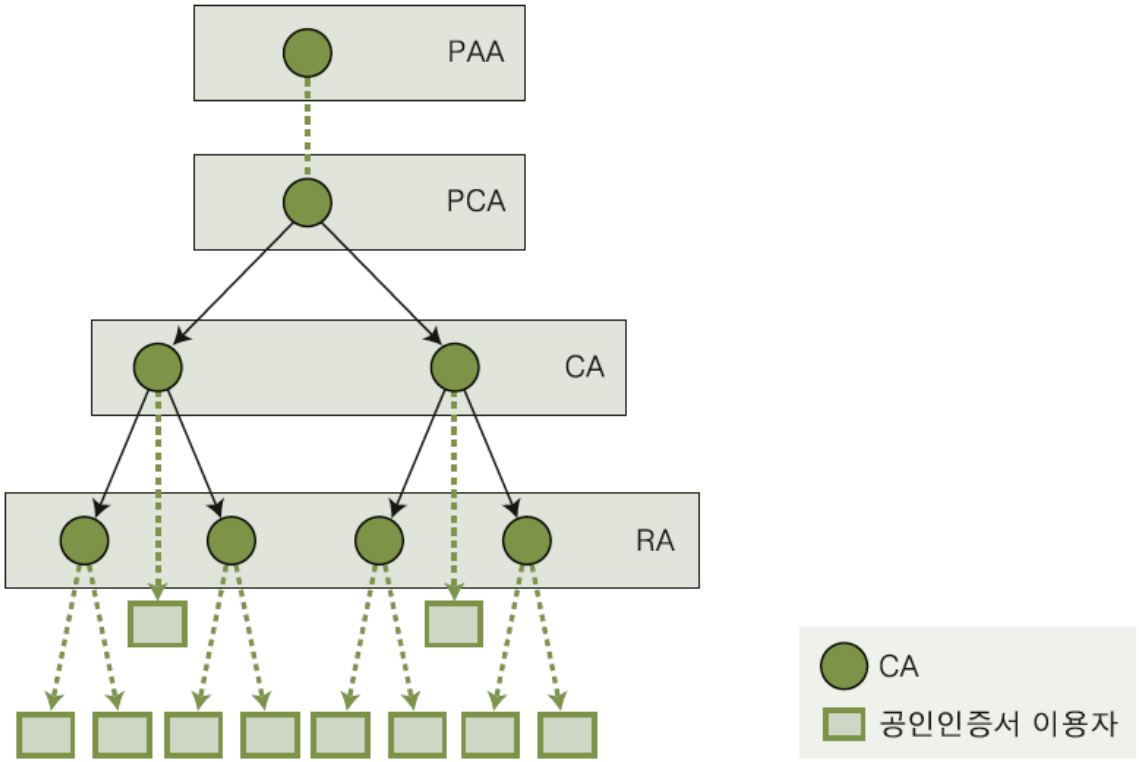
[그림 5] 동사무소에서 신분 확인을 위해 신분증 제시

- 공개키 기반 구조는 '인터넷에서 신분증을 검증해주는 관청'이라고 생각할 수 있음.
  - 가장 가까운 관청이 동사무소고, 그 동사무소들 위에 구청이 있고, 구청 위에 시청이 있고, 맨 위에 정부가 있는 것과 같음.
- 공개키 기반 구조하에 있는 사람은 어디에 가서도 자신의 인터넷상 신분을 CA(인증기관 : Certification Authority)와 공인인증서를 통해 증명할 수 있음.
  - CA가 일종의 동사무소고, 공인인증서가 주민등록증과 같은 신분증

02 공개키 기반 구조

공개키 기반 구조의 개념

- 공개키 기반 구조가 되기 위해서는 인증 정보를 일원화하여 호환성을 갖추고 있어야 하고 개인이 이를 쉽게 접근할 수 있어야 함.
  - 이를 위해 앞서 이야기한 동사무소, 구청, 시청, 정부와 같은 트리형 구조가 필요



[그림 6] 공인인증서 운영을 위한 계층 구조

02 공개키 기반 구조

공개키 기반 구조의 개념

- PAA(Policy Approval Authorities, 정책승인기관)
  - 공인인증서에 대한 정책을 결정하고 하위 기관의 정책을 승인하는 기관
  - 우리나라는 미래창조과학부가 담당

- PCA(Policy Certification Authorities, 정책인증기관)
  - RootCA를 발급하고 기본 정책을 수립하는 기관
  - 우리나라의 KISA(Korea Information Security Agency, 한국정보보호진흥원)가 여기에 해당
  - RootCA는 모든 인증서의 기초가 되는 인증서를 보유하고 있음.
  - 인증서에 포함된 공개키에 대응되는 개인키로 생성한 자체 서명 인증서를 사용

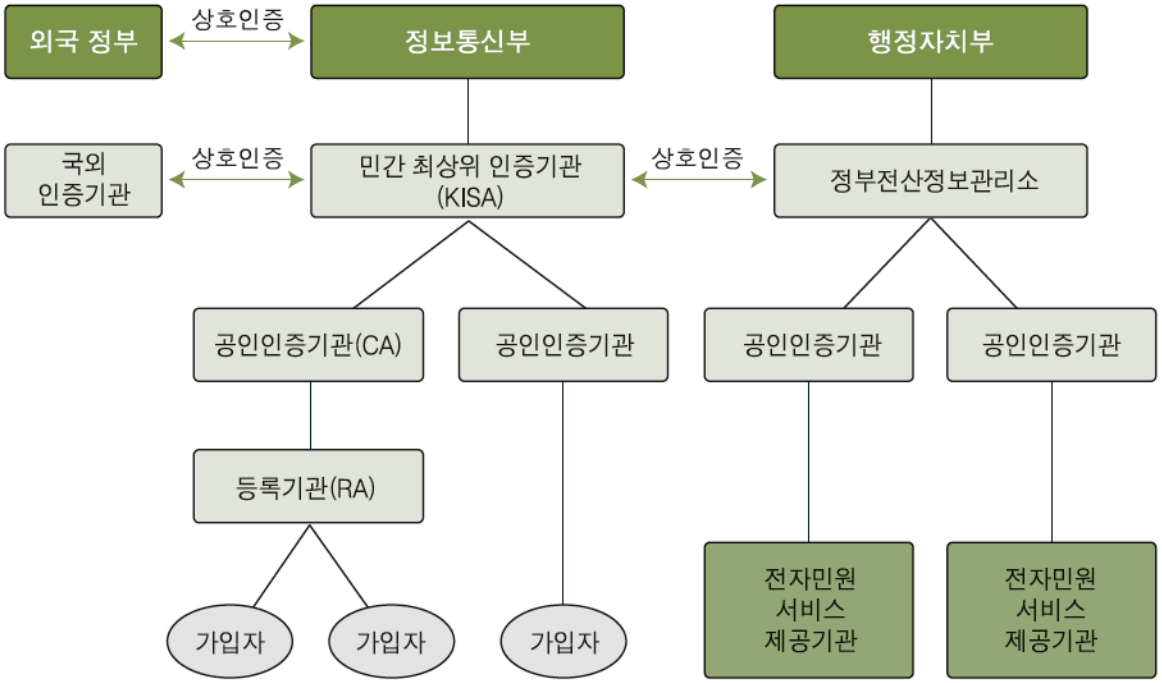
- CA(Certification Authority, 인증기관)
  - PCA의 하위 기관으로 인증서 발급과 취소 등의 실질적인 업무를 하는 기관
  - yessign(금융결제원), NCA(한국 전산원) 등이 이에 속하며, 상호 간 신뢰함.

- RA(Registration Authority, 등록기관)
  - 사용자의 신분을 확인하고 CA 간 인터페이스를 제공하는 기관

02 공개키 기반 구조

공개키 기반 구조의 개념

- 네트워크 구조 모델은 인증기관이 상호인증(crosscertification)을 통해 연결되어 있는 모델
- 상호인증이란 두 인증기관이 상대방의 공개키를 서로 인증해주는 인증서를 발급하여 사용하는 것
  - 이때 인증서를 상호인증서(cross-certificate)라 함.
- 일반적으로는 계층 구조와 네트워크 구조를 혼합해 사용

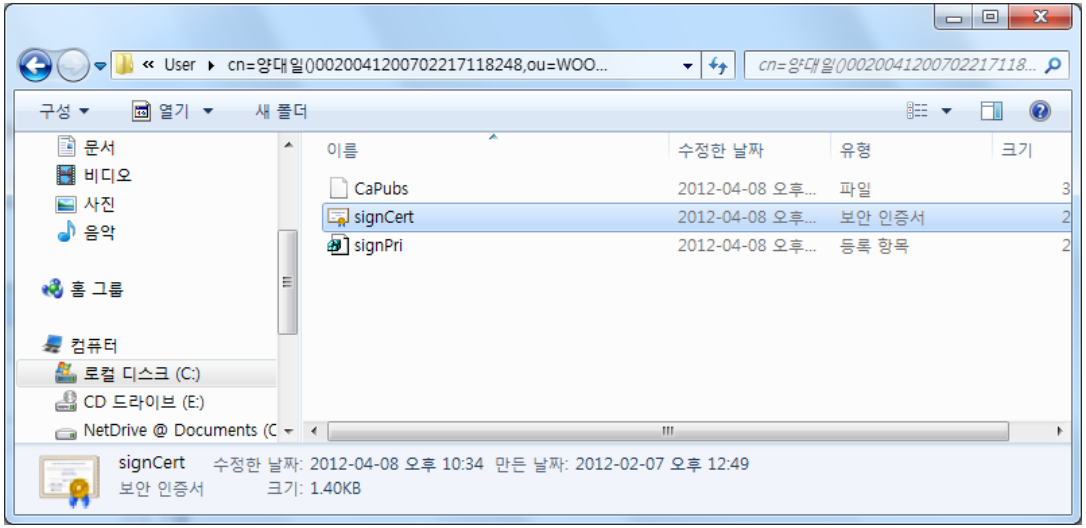


[그림 7] 국가와 기관 간 상호인증 구조

02 공개키 기반 구조

공인인증서에 대한 이해

- 공개키와 그것의 소유자를 연결시켜주는 전자문서
- 펠더(Kohnfelder)가 1978년에 처음 제안
- 공인인증서는 신뢰할 수 있는 인증기관(CA)이 전자서명하여 생성
- 오늘날 사용되는 대부분의 인증서는X.509 버전 3 표준을 따름.
  - 이 표준 이외에도 SPKI(Simple Public Key Infrasturcture) 인증서, PGP(Pretty Good Privacy) 인증서가 있음.
- 공인인증서는 인증서를 발급한 CA 이름을 기준으로 저장

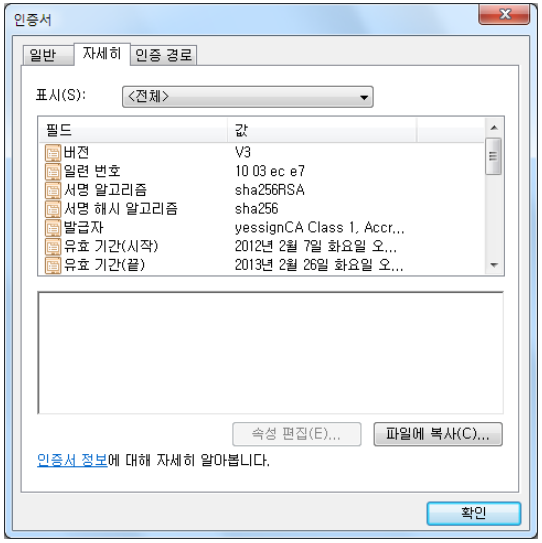


[그림 8] 인증서가 설치된 디렉터리

02 공개키 기반 구조

공인인증서에 대한 이해

- 인증서가 가지는 정보



[그림 9] 인증서 내용 열람

- **인증서의 기본 영역**
  - ❶ 버전 : 인증서의 형식 구분(우리가 사용하는 대부분의 공인인증서는 버전3)
  - ❷ 일련번호 : 인증서를 발급한 인증기관 내의 인증서 일련번호
  - ❸ 서명 알고리즘 : 인증서를 발급할 때 사용한 알고리즘
  - ❹ 발급자 : 인증서를 발급한 인증기관의 DN(Distinguish Name)
  - ❺ 유효 기간(시작, 끝) : 인증서를 사용할 수 있는 기간(시작일과 만료일을 기록하며 초 단위까지 표기됨).
  - ❻ 주체 : 인증서 소유자의 DN
  - ❼ 공개키 : 인증서의 모든 영역을 해시해서 인증기관의 개인키로 서명한 값

02 공개키 기반 구조

■ 공인인증서에 대한 이해

▪ 인증서의 확장 영역

❶ 기관 키 식별자 : 인증서를 확인할 때 사용할 인증기관 공개키의 유일 식별자

❷ 주체 키 식별자 : 인증서 소유자의 공개키에 대한 유일 식별자

❸ 주체 대체 이름 : 인증서 사용자의 이름 혹은 또 다른 별개의 이름에 대한 부가 정보로 사용자 ID, E-mail, IP 주소, DNS 이름 등을 표시(버전3에서는 x.500DN 이외에 하나의 대체 이름을 가질 수 있음)

❹ CRL 배포 지점 : 인증서의 폐기 여부를 확인하기 위한 인증서 폐기 목록(CRL)이 있는 위치

❺ 기관 정보 액세스

❻ 키 사용 용도 : 인증서에 포함된 공개키의 용도를 나타냄

❼ 인증서 정책

❽ 손도장 알고리즘

❾ 손도장

▪ 인증서의 특성

- 누구나 사용자의 인증서를 획득하고, 공개키를 획득할 수 있음.
- 인증기관 이외에는 인증서를 수정/발급할 수 없음.
- 같은 인증 구조 내의 사용자는 상호인증서 신뢰가 가능

02 공개키 기반 구조

공인인증서에 대한 이해

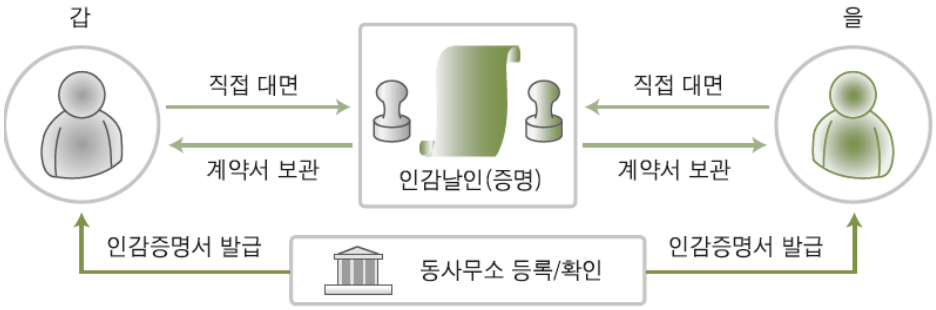
인증서의 폐기

- 폐기된 인증서의 사용에 따른 피해를 줄임.
- 폐기된 인증서의 사용에 따른 피해를 줄이기 위해 인증기관은 폐기된 인증서 목록을 주기적으로 발급.
  - 이를 인증서 폐기 목록(CRL, Certification Revocation List)이라 함. (X.509 표준에 정의되어 있음)
  - 이 목록도 인증서처럼 임의로 조작하거나 만들 수 없어야 함. 따라서 인증서처럼 인증기관이 전자서명을 하여 발급
- 인증서 폐기 목록은 보통 폐기된 인증서에 관한 정보만 유지
  - 이와 같은 접근 방법을 나쁜 목록(bad-list) 방법이라 함.
  - 반대로 좋은 목록(good-list) 방법도 있음.
  - 좋은 목록에서는 이 목록에 포함된 인증서만 사용
  - 나쁜 목록에서는 이 목록에 포함되지 않은 인증서만 사용해야 함.

03 전자서명과 전자봉투

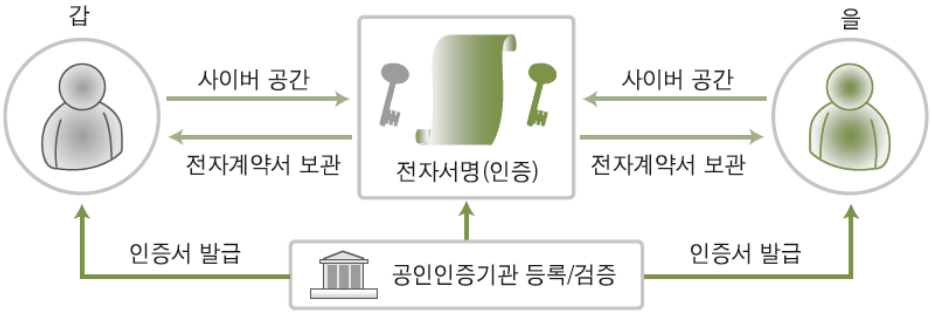
■ 전자서명

- 서명이란 서명한 사람의 신분을 집약적으로 증명하는 도구로 전자서명도 이와 비슷
- 우리나라의 전자서명법의 정의 : 전자서명이란 서명자가 해당 전자문서에 서명하였음을 나타내기 위해 전자문서에 첨부되거나 논리적으로 결합된 전자적 형태의 정보를 말한다.
- 계약을 할 때 사용하는 인감도장은 동사무소 등과 같은 공공기관에 등록하여 공증을 받은 것으로, 계약서 등의 날인에 사용.



[그림 10] **인감도장을 사용한 계약서 날인**

- 전자서명이 인감도장이 되고, 두 사람의 전자서명은 공인 인증기관에 등록되고 검증되어 사용.

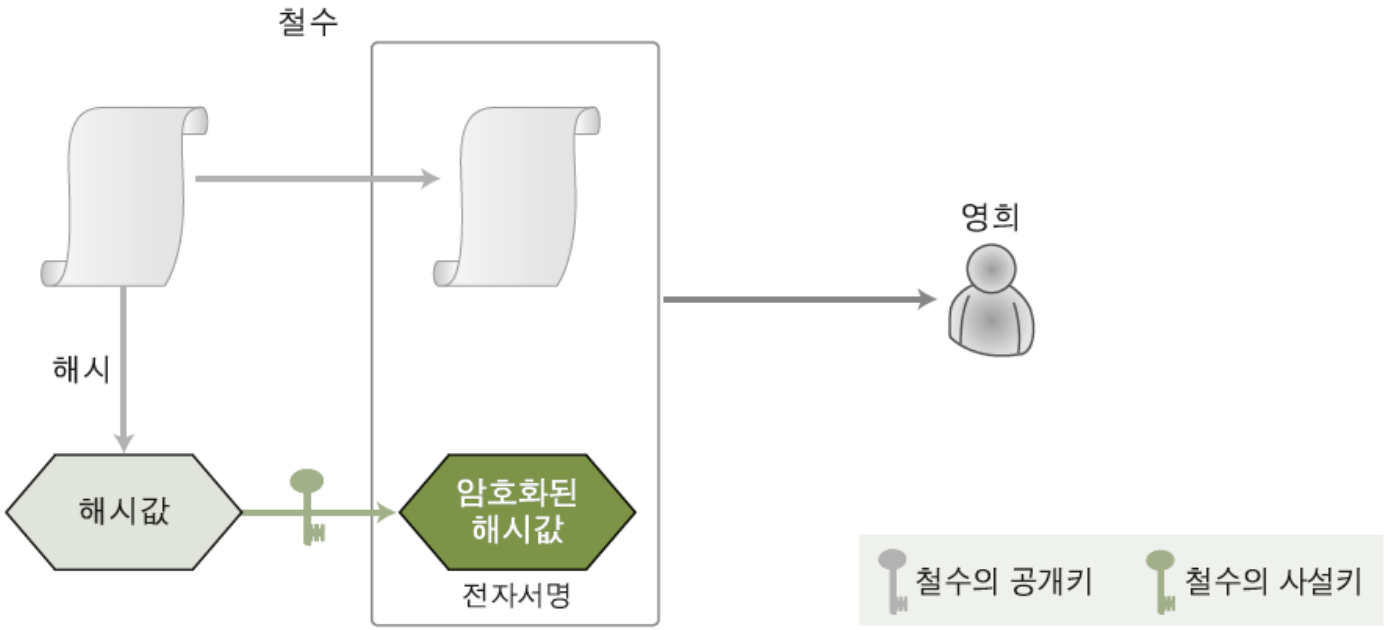


[그림 11] **전자서명을 사용한 인증**

03 전자서명과 전자봉투

■ 전자서명

- ▪ 전자서명의 원리
  - • 전자서명에서는 원본의 해시값을 구한 뒤, 그 해시값에 부인방지 기능을 부여하기 위해 공개키 방법을 사용
    - – 철수가 영희에게 편지를 보낼 때, 편지의 해시값을 구한 후 그 해시값을 자신의 사설키로 암호화하여 보냄.

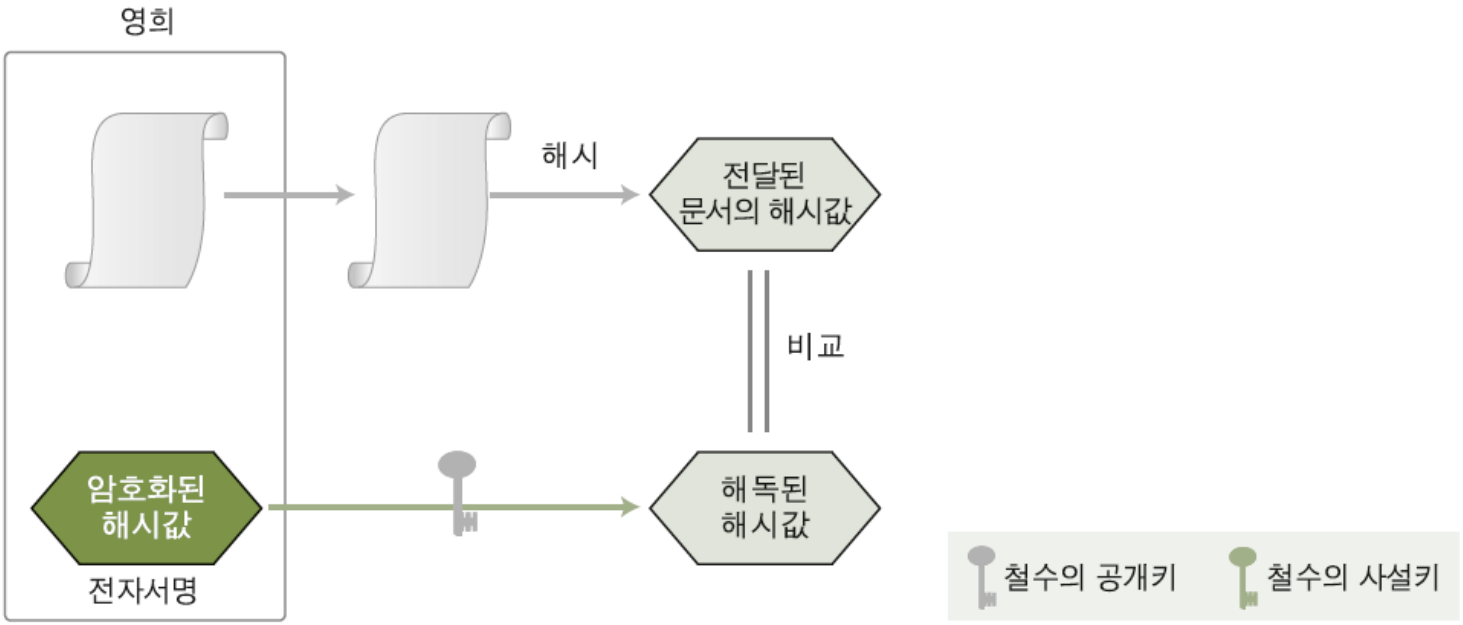


[그림 12] 전자서명의 생성

03 전자서명과 전자봉투

■ 전자서명

- ▪ 전자서명의 원리
  - • 영희는 철수의 공개키를 이용해 암호화된 해시값을 복호화하고, 원본 문서를 해시한 값과 비교
    - – 복호화한 해시값과 전달된 편지에서 구한 해시값이 일치하면 전달된 편지가 철수로부터 온 것이 맞고, 위조되지 않았음을 확신할 수 있음.



[그림 13] 전자서명을 이용한 전송 문서 확인

03 전자서명과 전자봉투

전자서명

- 전자서명이 제공하는 기능
  - 위조 불가(Unforgeable) : 서명자만이 서명문을 생성할 수 있음.
  - 인증(Authentication) : 서명문의 서명자를 확인
  - 재사용 불가(Not Reusable) : 서명문의 해시값을 전자서명에 이용하므로 한 번 생성된 서명을 다른 문서의 서
    명으로 사용할 수 없음.
  - 변경 불가(Unalterable) : 서명된 문서는 내용을 변경할 수 없기 때문에 데이터가 변조되지 않았음을 보장하는 무결성을 만족
  - 부인 방지(Non Repudiation) : 서명자가 나중에 서명한 사실을 부인할 수 없음.

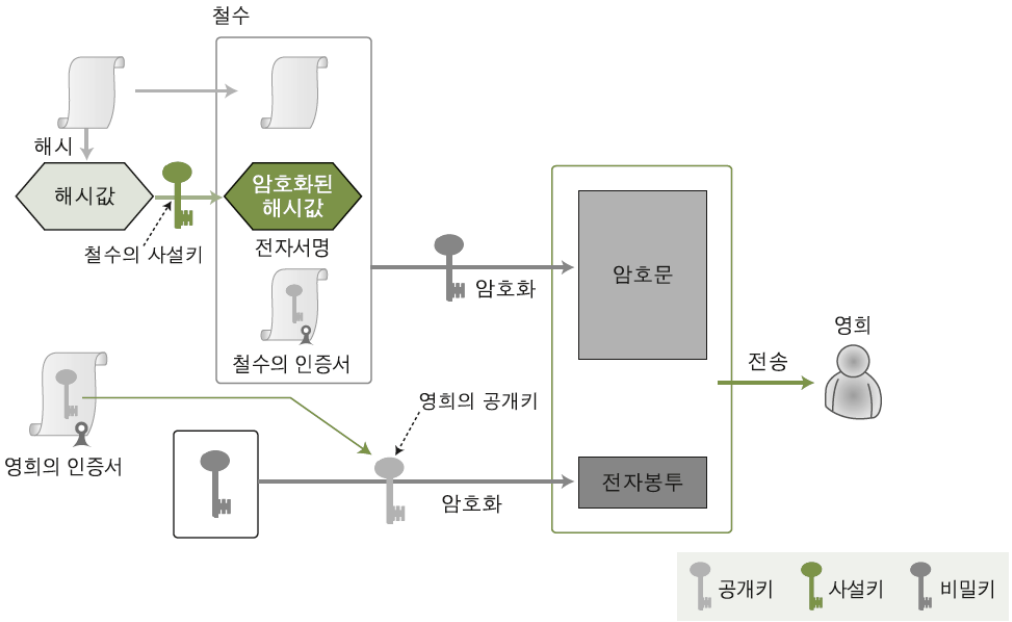
- 전자서명에 관련해 미국에는 1994년에 만들어진 DSS(Digital Signature Standard)가 있고, 이는 DSA(Digital Signature Algorithm)를 사용함.
  - DSA는 슈노어(Schnorr)와 엘가말(ElGamal)의 알고리즘을 기반으로 함.
  - 서명 생성이나 암호키 생성에서는 SHA.1을 이용하고 있음.

- 우리나라에서는 1996년에 개발된 KCDSA(Korean Certifice-based Digital Signature Algorithm)가 있음.
  - 현재 우리나라의 전자서명법에 따르면, 전자서명은 인터넷을 통해 전자문서를 교환할 때 일반 문서에서 쓰이는 인감도장과 법적으로 똑같은 효력을 지님.

03 전자서명과 전자봉투

전자봉투

- 전달하고자 하는 메시지를 암호화하여 한 사람을 통해서 보내고, 암호화 키는 다른 사람에게 가져가게 하는 것을 암호학적으로 구현한 것.
- 철수는 전자봉투를 사용하기 위해 우선 전자서명을 생성하고 전자서명과 원문, 그리고 자신의 공개키가 들어있는 인증서를 비밀키(DES 알고리즘 등에 사용되는 대칭키)를 사용하여 암호화함.
- 전자서명 세트와 인증서를 암호화한 비밀키를 영희의 공개키로 암호화
  - 이것이 전자봉투가 됨.
- 철수는 최종적으로 비밀키로 암호화한 결과와 비밀키가 암호화된 전자봉투를 영희에게 보냄.

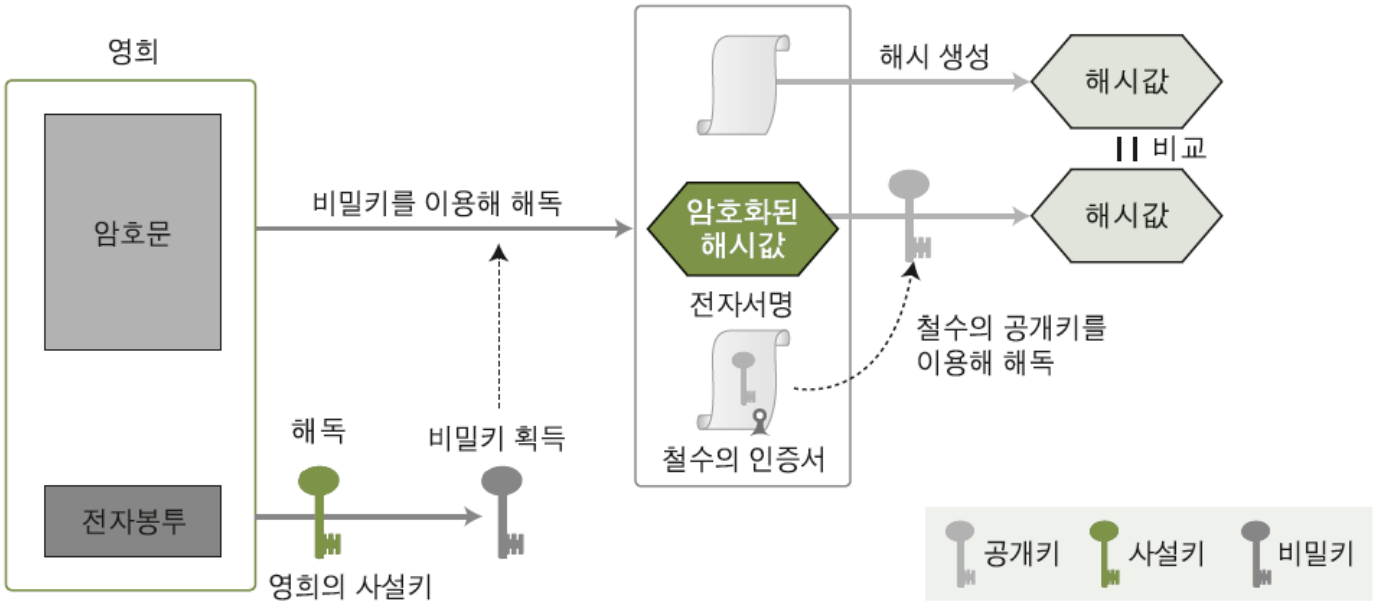


[그림 14] 전자봉투를 이용한 암호화 전송

03 전자서명과 전자봉투

전자봉투

- 전달받은 영희는 우선 전자봉투를 자신의 사설키로 복호화하여 비밀키를 획득
- 비밀키를 이용하여 전자서명과 평문, 철수의 인증서를 복호화(해독)
- 복호화한 인증서에서 철수의 공개키를 얻어 전자서명을 복호화한 후 이를 원문 해시 결과와 비교



[그림 15] 전자봉투의 복호화

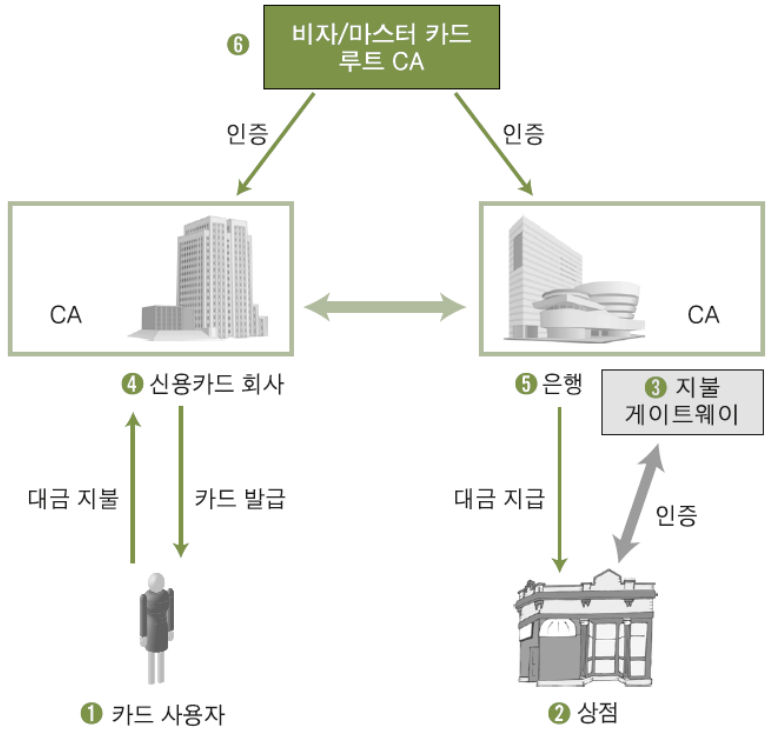
- 전자봉투는 기밀성, 무결성, 부인 방지를 모두 지원함.

04 전자결재

SET

- SET(Secure Electronic Protocol) : 1996년 비자(Visa)와 마스터(Master) 카드 회사의 합의에 의해 만들어진 프로토콜

- **SET의 구성**



[그림 16] SET의 구성

❶ 카드 사용자 : 신용카드를 소지한 사람으로 SET에 이용되는 인증서를 소유

❷ 상점 : 인터넷 쇼핑몰을 운영하며 SET를 이용하여 상품을 판매

❸ 지불 게이트웨이(PG: Payment Gateway) : 기존의 신용카드 지불 방식으로 은행과 거래 내역을 주고받음.

❹ 신용카드 회사(Issuer) : 사용자에게 신용카드를 발급하고, CA를 운영하여 사용자에게 인증서를 발급

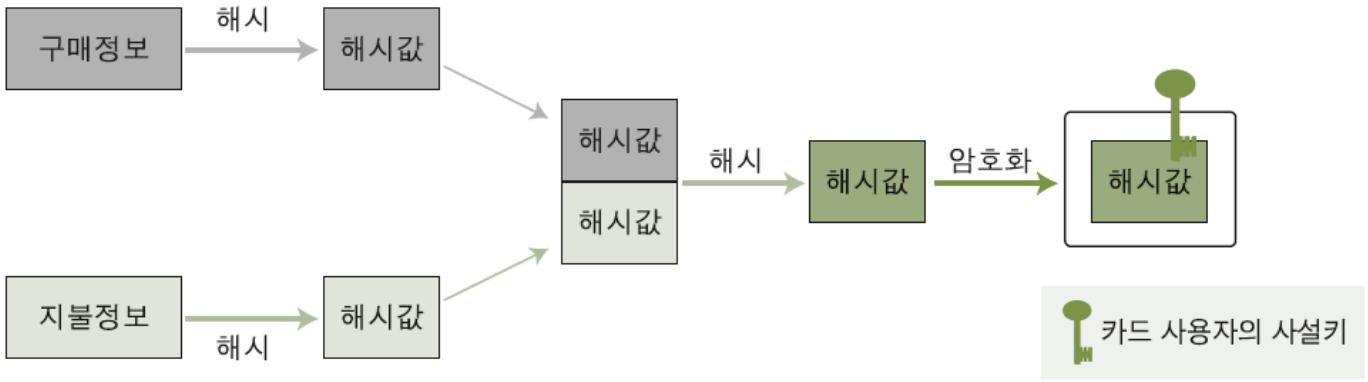
❺ 은행(Acquirer) : 상점의 계좌가 있으며 지불 게이트웨이를 운영함. CA를 운영하며 상점에 인증서를 발급

❻ 인증기관 : SET에 참여하는 모든 구성원의 정당성을 보장하는 루트(Root) CA

04 전자결재

SET

SET의 지불 과정



[그림 17] 이중 서명의 기본 동작

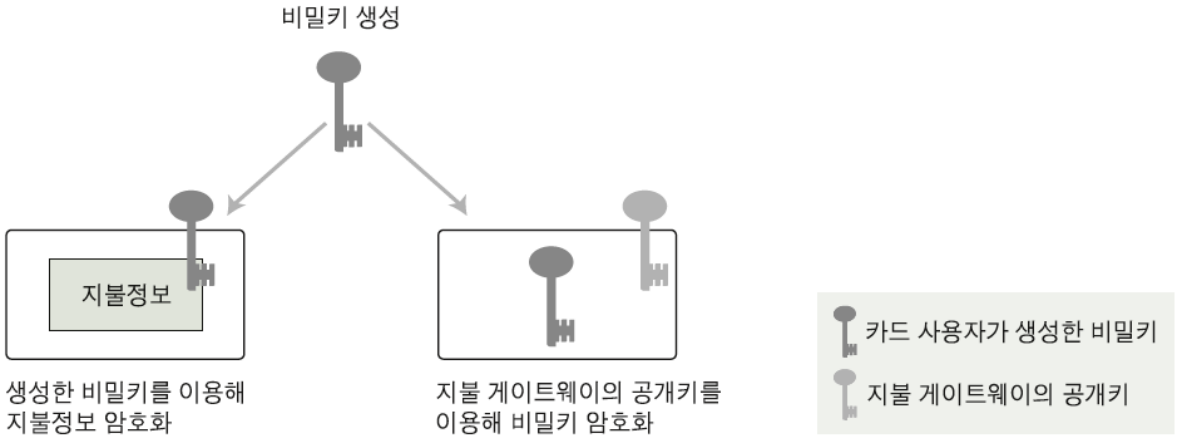
- 이중 서명의 생성
  - 카드 사용자가 구매정보와 지불정보를 각각 해시
  - 두 해시값을 합한 뒤 다시 해시
  - 최종 해시값을 카드 사용자의 사설키로 암호화(서명)
- 이중 서명의 목적
  - 상점이 카드 사용자의 계좌번호 같은 지불정보를 모르게 함.
  - 상점에 대금을 지불하는 은행이 카드 사용자가 상점에서 산 물건이 무엇인지 모르면서 상점이 요구한 결제 대금이 정확한지 확인할 수 있게 하는 데 있음.

04 전자결재

SET

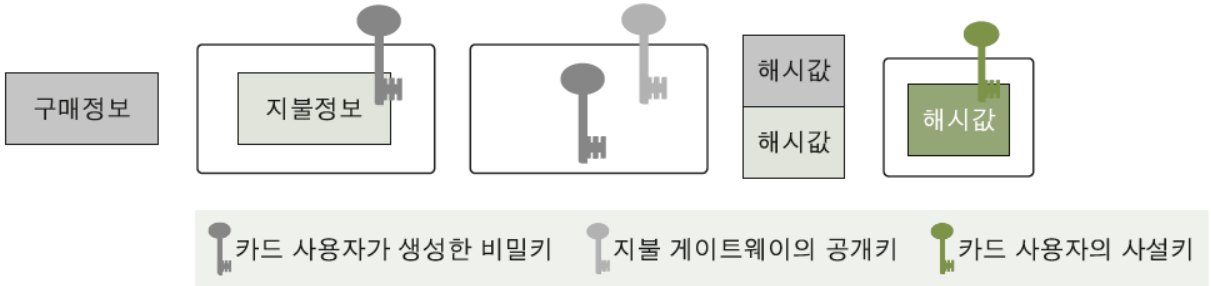
SET의 지불 과정

- 비밀키(대칭키) 생성
  - 비밀키를 사용해 지불정보를 암호화하고, 비밀키는 은행이 운영하는 지불 게이트웨이의 공개키로 암호화.



[그림 18] **비밀키의 생성**

- 결제를 위한 데이터 전송



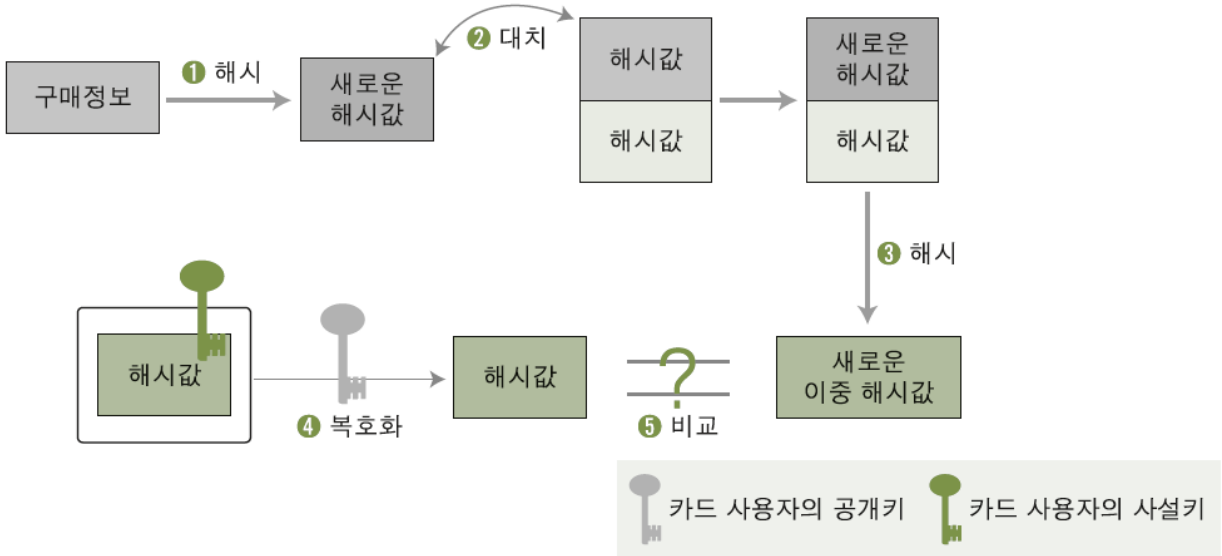
[그림 19] **결제를 위해 카드 사용자가 상점에 전송하는 데이터**

04 전자결재

SET

SET의 지불 과정

- 카드 사용자로부터 구매정보 확인
- 상점은 카드 사용자가 신청한 물건에 대한 구매정보의 해시를 구하고(❶), 카드 사용자가 보내온 한 쌍의 해시값을 새로 구한 해시로 대치시킨 뒤(❷), 새로운 이중 해시를 구함(❸). 그 후 카드 사용자의 사설키로 암호화된 해시값을 복호화하여(❹) 이를 새로 구한 이중 해시값과 비교(❺)
- 그런 다음 카드 사용자가 보내온 구매정보가 그 카드 사용자의 것인지 또는 구매정보가 변조되지 않았는지 확인



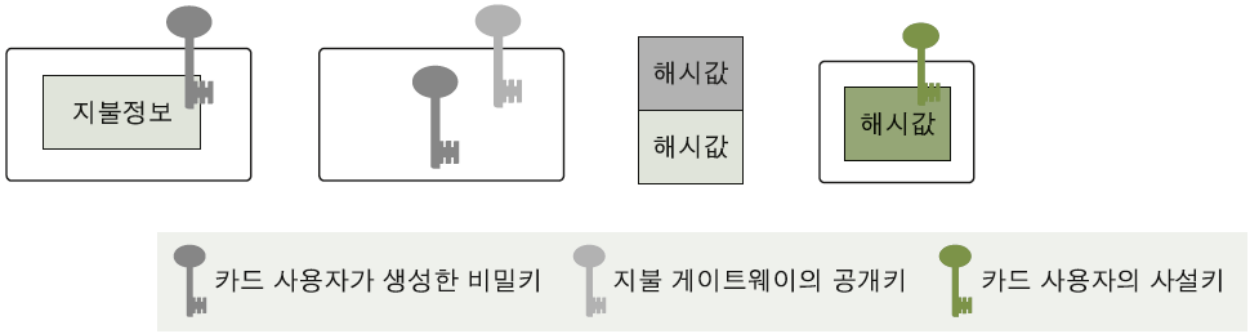
[그림 20] 이중 해시값을 이용한 구매정보 확인

04 전자결재

SET

SET의 지불 과정

- 상점은 지불 게이트웨이로 지불 정보 전송



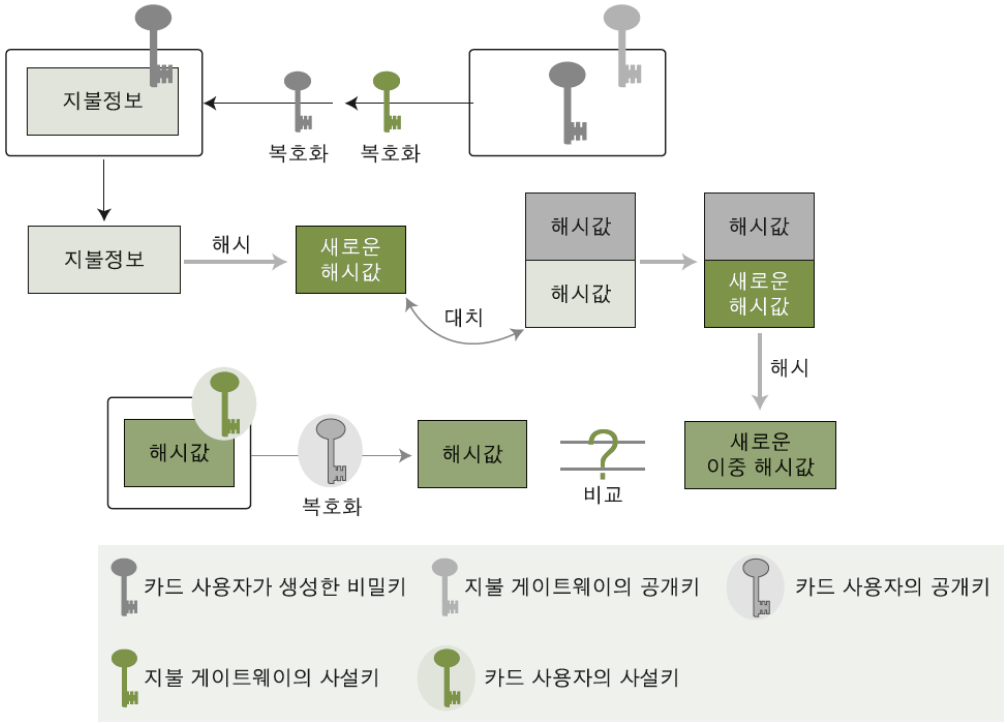
[그림 21] 상점이 지불 게이트웨이로 보내는 데이터

04 전자결재

SET

SET의 지불 과정

- 지불 정보 확인
  - 지불 게이트웨이는 자신의 사설키로 비밀키를 복호화하여 지불정보를 확인
  - 상점이 한 것처럼 받은 지불정보를 해시한 값으로 한 쌍의 해시값을 대치하여 이중 해시값을 비교하고, 지불정보의 변조 여부를 확인한 뒤 상점에 대금을 지불.



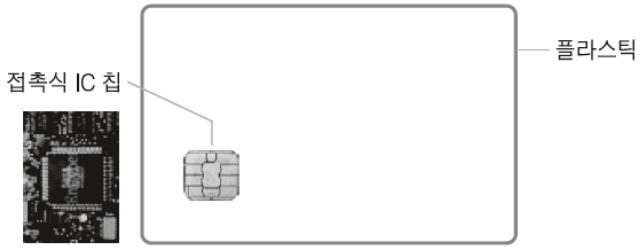
[그림 22] 지불정보의 확인

04 전자결재

■ 스마트카드

▪ 접촉식 카드

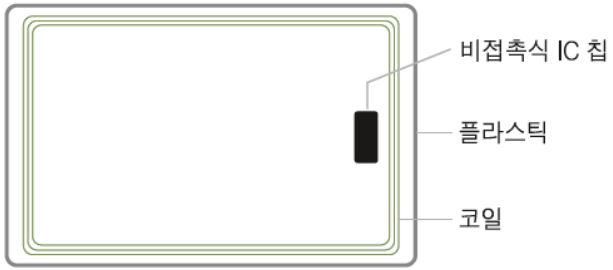
- 접촉식 스마트카드 리더기와 스마트카드의 접촉부(CHIP) 사이의 물리적 접촉에 의해 작동하는 스마트카드
- 접촉식 카드는 접점의 잦은 접촉으로 인해 전기적 충격이나 손상이 생길 우려가 있음.
- 보안이 중요한 많은 DATA를 처리하는 거래 인증, 전자서명 등의 응용에 적합



[그림 24] 접촉식 카드의 구조

▪ 비접촉식 카드

- 구리선을 이용하여 무선 주파수 파장을 전력으로 전환하는 방식으로 구동시켜 스마트카드 리더기와 통신하는 카드
- 처리 시간에 제한을 받는 교통, 유통 등에 적합



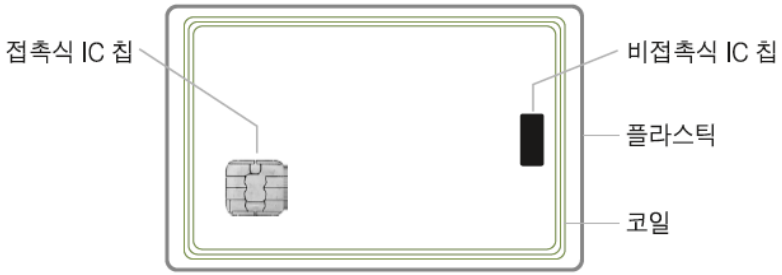
[그림 25] 비접촉식 카드의 구조

04 전자결재

■ 스마트카드

▪ 하이브리드 카드

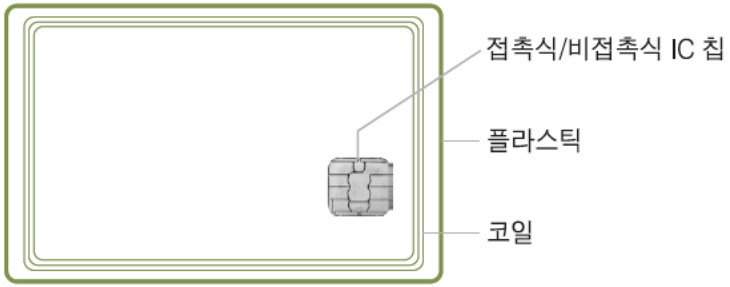
- 하나의 카드 안에 접촉식 카드와 비접촉식 카드가 별도로 존재하는 하이브리드(Hybrid) 형식의 카드
- 하드웨어와 소프트웨어 역시 별도로 존재



[그림 26] 하이브리드 카드의 구조

▪ 콤비 카드

- 접촉식 카드와 비접촉식 카드가 공유할 수 있는 부분을 상호 공유하는 스마트카드



[그림 27] 콤비 카드의 구조

04 전자결재

스마트 카드

- SIM 카드
  - 가입자 식별 모듈(Subscriber Identification Module)을 구현한 IC 카드
  - GSM 단말기의 필수 요소
  - 3세대 이동통신 단말기에서는 USIM(Universal Subscriber Identity Module, 범용 사용자 식별 모듈)으로 표준이 확장됨.
  - 4세대 이동통신인 LTE에서도 일부 수정되어 사용되고 있음.

[그림 28] USIM

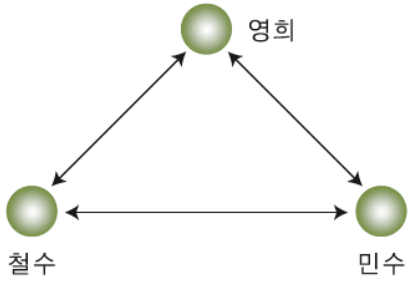
05 암호화 통신

전자우편의 암호화

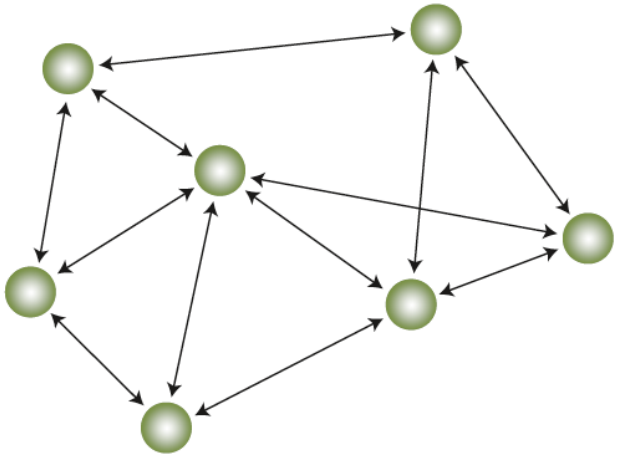
PGP[pretty good privacy]

- 필 치머만(Phil Zimmermann)이 독자적으로 개발
- PGP를 사용하는 사람들끼리의 신뢰 관계를 통해 인증
  - 철수가 영희, 민수와 PGP를 통해 서로 신뢰하는 관계라면, 영희와 민수도 철수를 통해 서로를 신뢰하게 됨.
  - 이는 공인인증서에서 살펴본 상호인증서를 통한 네트워크 구조와 유사.
  - PGP는 이런 상호인증을 통해 많은 인터넷 사용자가 서로를 인증하여 그물망과 같은 인증구조를 가지게 됨.

[그림 33] 필 치머만



[그림 34] PGP 상호인증 예

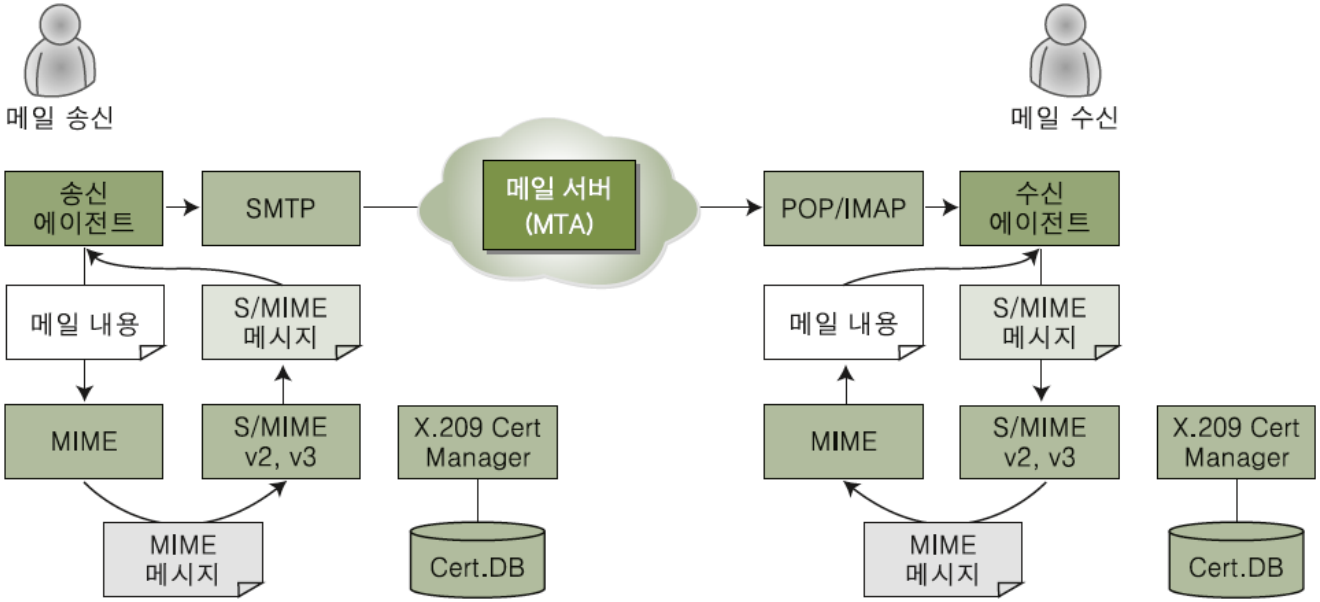


[그림 35] 인터넷에서의 PGP 상호인증

05 암호화 통신

전자우편의 암호화

- **S/MIME**[Security Services for MIME, Security Services for Multipurpose Internet Mail Extension]
  - S/MIME(Secure MIME)은 인증서 서비스를 통하여 암호화되는 메일 서비스를 제공
  - S/MIME 관련 프로그램을 설치하면 대부분 자동으로 이루어짐.
  - S/MIME은 아직까지 널리 쓰이지 않으나 회사에서 그룹웨어를 사용할 때 이와 매우 비슷한 형태의 암호화 메일을 제공하는 경우가 많음.



[그림 36] S/MIME의 동작

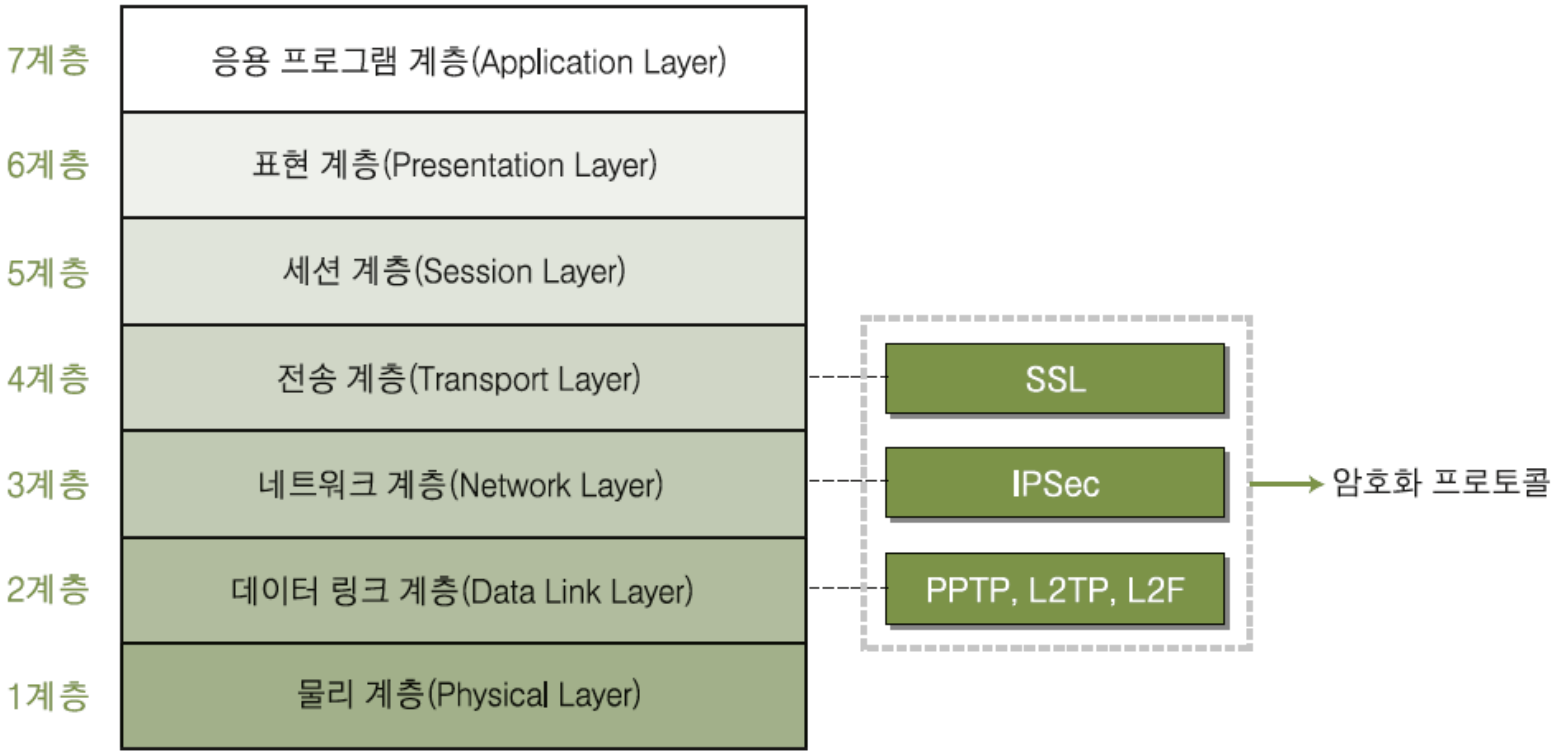
05 암호화 통신

전자우편의 암호화

- **PEM [privacy enhanced mail]**
  - IETF(Internet Engineering Task Force)에서 채택
  - 높은 보안성을 가지고 있음.
    - 하지만 구현의 복잡성 등의 이유로 널리 쓰이지는 않음.

05 암호화 통신

■ 네트워크 암호화



[그림 37] OSI 계층별 암호화 프로토콜

05 암호화 통신

■ 네트워크 암호화

▪ 데이터 링크 계층의 암호화 프로토콜

- **PPTP** [point-to-point tunneling protocol]
  - 마이크로소프트가 제안한 VPN 프로토콜로, PPP(Point-to-Point Protocol)에 기초.
  - PPP는 두 대의 컴퓨터가 직렬 인터페이스를 이용하여 통신할 때 사용
    - » 특히 전화선을 통해 서버에 연결하는 PC에서 자주 사용됨.
- **L2TP** [layer 2 tunneling protocol]
  - 시스코가 제안한 L2F(Layer 2 Forwarding)와 PPTP가 결합된 프로토콜
  - PPTP와 L2TP는 모두 PPP 트래픽을 암호화
    - » IP, IPX,NetBEUI, AppleTalk 등의 다양한 상위 로컬 네트워크 프로토콜을 사용할 수 있음.
  - 둘다 사용자 인증(PAP, CHAP, MS-CHAP,EAP)이나 데이터 암호화/압축(CCP, ECP) 등의 보안 기능을 PPP에서 제공하는 것을 사용.

[표 2] PPTP와 L2TP 프로토콜의 비교

	PPTP	L2TP
네트워크	통신하기 위해 양단의 네트워크가 IP를 기반으로 해야 한다.	프레임 릴레이(Frame Realy), ATM 등에서도 사용할 수 있다.
터널링	두 시스템 사이에 하나의 터널만 지원한다.	여러 개의 터널을 허용하여 QoS(Quality of Service)에 따라 서로 다른 터널을 이용할 수 있다.
압축 및 인증	해당 기능 없다.	헤더 압축 및 터널에 대한 인증 기능을 제공한다.

05 암호화 통신

■ 네트워크 암호화

▪ 네트워크 계층의 암호화 프로토콜

- **IPSec [IP security protocol]**
  - 3계층의 암호화 프로토콜
  - IP 스푸핑이나 스니핑 공격에 대한 대응 방안이 될 수 있음.
  - 주요 기능은 AH (Authentication Header)를 이용한 인증, ESP(Encapsulation Security Payload)를 이용한 기밀성, IKE(Internet Key Exchange)를 이용한 비밀키 교환임.

[표 3] IPSec 프로토콜의 기능

기능	특징
AH(Authentication Header)	•전송 도중에 데이터가 변조되었는지 확인할 수 있도록 데이터의 무결성을 검사한다. • 데이터를 스니핑한 뒤 해당 데이터를 다시 보내는 재생 공격(Replay Attack)을 막을 수 있다.
ESP(Encapsulating Security Payload)	•메시지의 암호화를 제공한다. • ESP에서 사용하는 암호화 알고리즘으로는 DESCBC, 3DES, RC5, IDEA, 3IDEA, CAST, blowfish가 있다.
IKE(Internet Key Exchange)	• ISAKMP(Internet Security Association and Key Management Protocol), SKEME, Oakley 알고리즘의 조합으로, 두 컴퓨터 간의 보안 연결(SA: Seucrity Association)을 설정한다. • IPSec에서는 IKE를 이용하여 연결이 성공하면 8시간 동안 SA를 유지하므로, 8시간이 넘으면 SA를 다시 설정해야 한다.

05 암호화 통신

네트워크 암호화

전송 계층의 암호화 프로토콜

- **SSL** [Secure Sockets Layer]
  - 네스케이프가 개발
  - 40비트와 128비트의 키를 가진 암호화 통신을 할 수 있게 해줌.
  - L2TP나 IPSec보다 상위 수준에서 암호화 통신기능을 제공
    - » 보통 4계층(전송 계층)과 5계층(세션 계층) 사이의 프로토콜이라 함.
  - SSL의 기능은 크게 서버 인증, 클라이언트 인증, 암호화 세션임.
  - 암호화된 통신은 40비트와 128비트의 암호화 세션을 형성
    - » 국내의 많은 사이트가 아직 40비트 암호화를 제공하는 모듈을 사용
  - 서버 인증은 클라이언트가 자신이 신뢰할만한 서버에 접속을 시도하고 있는지를 확인하는 클라이언트가 공개키 기술을 이용하여 서버의 인증서가 신뢰된 CA에서발행된 것인지를 확인
  - 서버는 클라이언트의 인증서를 확인하여 클라언트가 서버에 접속할 자격이 있는지를 확인할 수 있음.

[그림 38] OSI에서의 SSL의 동작 위치

06 컨텐츠 보안

스테가노그래피

- 워터마크와 비슷하지만 '저작권 보호'보다는 '정보를 은밀하게 전달'하기 위한 목적이 더 큼.

워터마크

- 편지지의 제작사를 표시하기 위해 편지지에 투명무늬를 희미하게 프린트한 것을 워터마크라고 부른데서 유래
  - 종이로 출력해 판매되는 IT 관련 문서의 페이지 전체에 옅은 색으로 소유권을 가진 회사의 로고를 표시한 것이 이에 해당.
  - 영상이나 오디오 파일에도 이런 데이터를 삽입
- 워터마크는 저작물의 사용자가 알아볼 수 있게 표시되기도 하지만 해당 저작물이 조작되지 않도록 인지할 수 없는 방식으로 표시되기도 함.

Q & A