제7장 암호 통제

암호화의 역사

- 암호화 개요
  - 파일이 노출되는 상황에서도 파일이 가지고 있는 정보를 노출시키지 않을 수 있는 마지막 정보보호 수단의 사용
  - 특히, 통신의 경우 옥외를 통하는 유무선 통신망을 통하여 파일이 전송되기 때문에 통신망에 대한 도청이 언제든지 일어날 수 있기 때문에 암호화가 필수적임
- 암호화의 역사
  - 시저 암호(Caesar cipher)
  - 2차 세계대전시: ENIGMA(독일), M-209(미국)
  - DES (Data Encryption Standard) : NIST에서 채택한 미국 표준
  - 공개키 암호화
  - 전자서명
  - 공개키 기반구조
- 고전 암호화 기법
  - 알파벳의 위치를 바꾸는 전치 (transposition)
  - 다른 부호로 대치하는 치환 (substitution)

암호화 개요

- 암호화란?
  - 누구나 읽을 수 있는 평문(plaintext)을 암호문(ciphertext)으로 바꾸어서 전송하거나 저장하는 방법
  - 평문은 적절한 알고리즘(algorithm)을 통하여 암호문으로 변환됨
  - 각 알고리즘은 키(key)라고 불리는 하나의 모수(parameter)를 가지고 있으며, 알고리즘의 종류에 따라 사용되는 키도 다름
  - 평문을 암호문으로 바꾸는 과정을 암호화(encryption)라고 하고, 암호문을 평문으로 바꾸는 과정을 복호화(decryption)라고 함
  - 과거에 컴퓨터가 개발되기 이전에는 알고리즘과 키를 모두 비밀로 유지함으로써 암호문의 비밀을 유지하는 방법을 사용
  - 컴퓨터가 사용되는 현대의 암호화에서는 키의 비밀성만을 통하여 암호문의 비밀성을 유지하며 알고리즘의 비밀성은 보유하지 않음
  - 현대의 암호화에서는 가능한 키의 개수가 충분히 커서 모든 가능한 키를 대입해보는 전수조사 해독법이 현실적으로 가능하지 않아야 함

암호 기술

▣ 암호 기술

- ◈ 정보통신의 발달로 컴퓨터의 의존도가 높아짐에 따라 발전
- ◈ 지식정보사회에서 야기될 수 있는 정보보호 문제를 해결
- ◈ 암호 기술의 두 가지 측면
  - 암호 알고리즘-암·복호화 하는 방식
  - 암호 프로토콜-암호기술을 바탕으로 제공되는 서비스

암호 기술

<table>
<tr><th colspan="4">암호 기술</th></tr>
<tr><th colspan="2">암호 알고리즘</th><th colspan="2">암호 프로토콜</th></tr>
<tr><th>대칭키</th><th>공개키(비대칭키)</th><th>단순</th><th>고급</th></tr>
<tr><td>•블록암호시스템
(DES)
•스트림암호시스템</td><td>•(결정론적)공개키
•확률론적 공개키
.
.
.</td><td>•사용자 인증
•메시지 인증
•전자서명
•부인 방지
•시점 확인
•키 분배와 멀티캐스팅</td><td>•전자지불/화폐(은닉 서명과 익명성)
•전자선거
•전자계약
•전자입찰
•전자공증
•지적재산권 보호시스템
•영지식 증명
•안전한 계산(비밀분산과 장애허용)</td></tr>
</table>

[그림 1] 현대 암호학의 분류

암호 기술

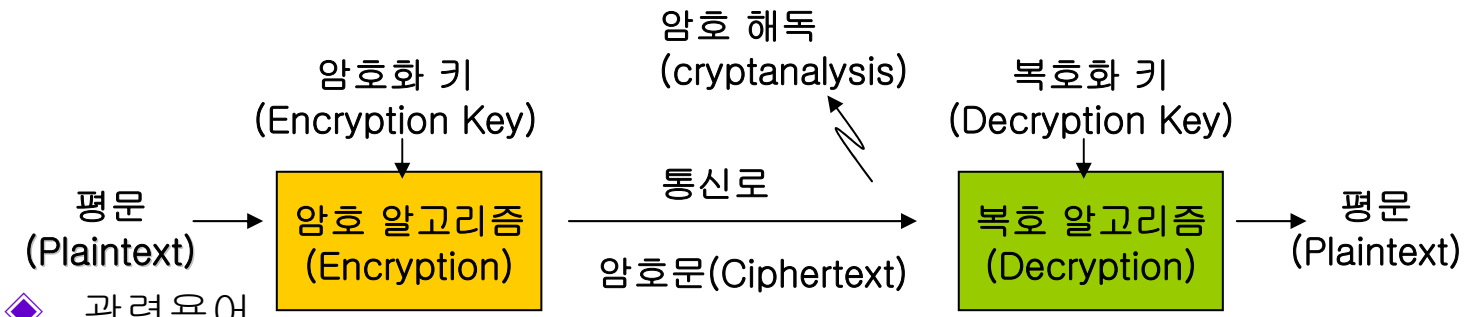
Mod연산 ▣나눗셈 연산으로, 어떤 값을 나누었을 때 나머지 값을 도출 ◈ 예) 8 mod3 = 2	소인수 분해 ▣어떤 값의 약수(인수)중에서 소수를 찾는 연산 ◈ 소수:1과 자신만을 약수로 갖는 수 ◈ 합성수:1 이외의 소수가 아닌 정수
이산대수 문제 ▣ $Y=a^x \pmod p$에서의 이산대수 ▣ $x=\log_a Y \pmod p$을 구하는 문제 ▣ p가 매우 큰 소수일 때, 현실적으로 풀기 어려움 ◈ 키 분배에 적용	해쉬 함수 ▣일방향성-해쉬 함수의 결과 값H(X)를 이용하여 X값을 구할 수 없음 ▣일정한 길이의 출력으로 압축 ◈ 암호 프로토콜에 적용

[그림 2] 암호 기술에 등장하는 수학적 배경

암호 기술의 기능

1. 비밀성 기능(암호화 기술)

- 정보통신망에서 전송되는 데이터의 불법적인 노출을 방지하는 기능
- 메시지를 제3자가 알 수 없게 변형, 또는 암호화된 통신문을 복호화 하는 원리, 수단, 방법 등을 다루는 기술 또는 과학



- 관련용어

[그림 3] 암호의 비밀성 기능의 원리

평문(plaintext)	송신자가 수신자에게 보내고자 하는 보통의 메시지
암호화(encryption)	평문을 제3자가 이해할 수 없는 형태의 암호문으로 변환시키는 조작
복호화(decryption)	암호문을 본래의 평문으로 바꾸는 조작
암호화/복호화 키(key)	암호/복호 알고리즘에 의해 평문/암호문의 변환을 제어하는 파라미터
암호 해독(cryptanalysis)	부당한 제3자가 다른 수단을 통해 평문을 알아내는 것
암호/복호 알고리즘	암호화/복호화 방식

암호 기술의 기능

2. 인증 기능(기본적인 암호 프로토콜 기술)

◈ 각종 정보보호 문제를 해결하는 기능
- 통신하는 사람 간의 신분확인 문제
- 전송되는 전자문서의 위·변조 방지 문제
- 전자적 행위에 대한 사후 부인 방지 문제
- 계약시간을 확인해주는 시점확인(time-stamp) 문제 등

◈ 분류
- **메시지 인증**
- **사용자 인증**
  - ✓ **개인식별**
- **디지털 서명**

- **거래 주체에 대한 인증** – 송신자, 수신자, 중재자가 필요할 수도 있음
- **거래 사실에 대한 인증** – 언제, 어디서.. (예: log)
- **거래 대상에 대한 인증** – 서비스, 제품

암호 알고리즘

구분	대칭키 암호 방식	공개키 암호 방식
키의 관계	암호화 키=복호화 키	암호화 키≠복호화 키
키 관리	암호화 키, 복호화 키 비밀	복호화 키 비밀
대표적인 예	DES	RSA
비밀키의 수 (사용자가 n명)	n(n-1)/2	n
암호키 분배 필요성	필요	불필요
안전한 인증	곤란	용이함
암호화 속도	고속	저속

[그림 4] 대칭키· 공개키 암호 알고리즘

블럭 암호와 스트림 암호

■ 블럭 암호

◈ 암호화와 복호화에 적용되는 키가 동일한가 또는 상이한가에 따라서 대칭형 암호와 공개키 암호로 분류됨

◈ n 비트의 평문 블록을 k 비트의 매개변수(key)를 이용해서 n비트의 암호문 블록으로 매핑시키는 함수

◈ 블럭화 과정-평문을 8문자씩 나누는 과정

(문자, 기호, 숫자는 ASCII 코드로 변환 시 8비트이기 때문)

■ 스트림 암호시스템

◈암호화의 단위가 비트 또는 문자(character)임

◈평문과 키를 이진수열로 표현하여 결합

◈ 배타적 논리합의 연산(exclusive-OR)으로 이진수열 결합

-서로 다른 비트가 결합되면 1, 같은 비트 결합일 때는 0의 값 도출

예) **0101 ⊕ 0111 = 0010**

대칭키 암호 알고리즘

▣ 대칭키 암호 알고리즘의 종류

◈ 치환(substitution)

– Julius Caesar의 Caesar 치환 암호

-알파벳에 순서를 두어 키만큼 해당문자의 위치를 옮기는 암호

-암호화 : **E(m) = E(m+ k) mod 26**

(단, m은 평문의 영문자에 대응하는 0과 25 사이의 정수)

-복호화 : **D(c)=D(c−k) mod 26**

(단, c는 암호문의 영문자에 대응하는 0과 25 사이의 정수)

m	A	B	C	D	E	F	G	H	I	J	K	L	M
순서	0	1	2	3	4	5	6	7	8	9	10	11	12

m	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
순서	13	14	15	16	17	18	19	20	21	22	23	24	25

[그림 5] Caesar 치환 암호

대칭키 암호 알고리즘

– 치환의 예

비밀 통신을 하고자 하는 송신자 A가 암·복호화 키 K=3을 가지고
평문 'ILOVEYOU'을 암호화 하여 수신자 B에게 전송하는 경우

· 암호화 (**E(m) = E(m+ 3) mod 26**)

평문: *I LOVE YOU* → 암호문: *L ORYH BRX*

· 복호화 (**D(c) = D(c−3) mod 26**에 대입)

암호문: *L ORYH BRX* → 평문: *I LOVE YOU*

– 치환 공격방법–알파벳의 빈도수 이용

- 알파벳 E는 12.75%, Z는 0.0009% 정도의 비율을 갖는데, 만약 암호문 상의 특정문자가 12.75%의 빈도수를 보이면 이 암호문자의 평문은 E일 확률이 높음

대칭키 암호 알고리즘

◈ 전치(permutation)

- 평문을 재배열하는 방식
- 평문 문장을 키의 길이에 따라 일정 간격으로 나누고, 이를 키의 재배열 순서에 따라 재배치 함

예) MATHEMATICS의 재배치

· 평문을 일정 간격으로 나눔 (d=4, 4문자씩 배열)

→ MATH EMAT ICSW (모자라는 문자는 임의로 채워줌, W)

1234 1234 1234

· 특정순서로 문자를 재배치 (키의 재배열 순서: 2 4 3 1)

→ AHTM MTAE CWSI

2431 2431 2431

* 치환과 전치는 가장 단순한 대칭키 암호 알고리즘으로 복호화는 암호화의 단순한 역조작으로 가능

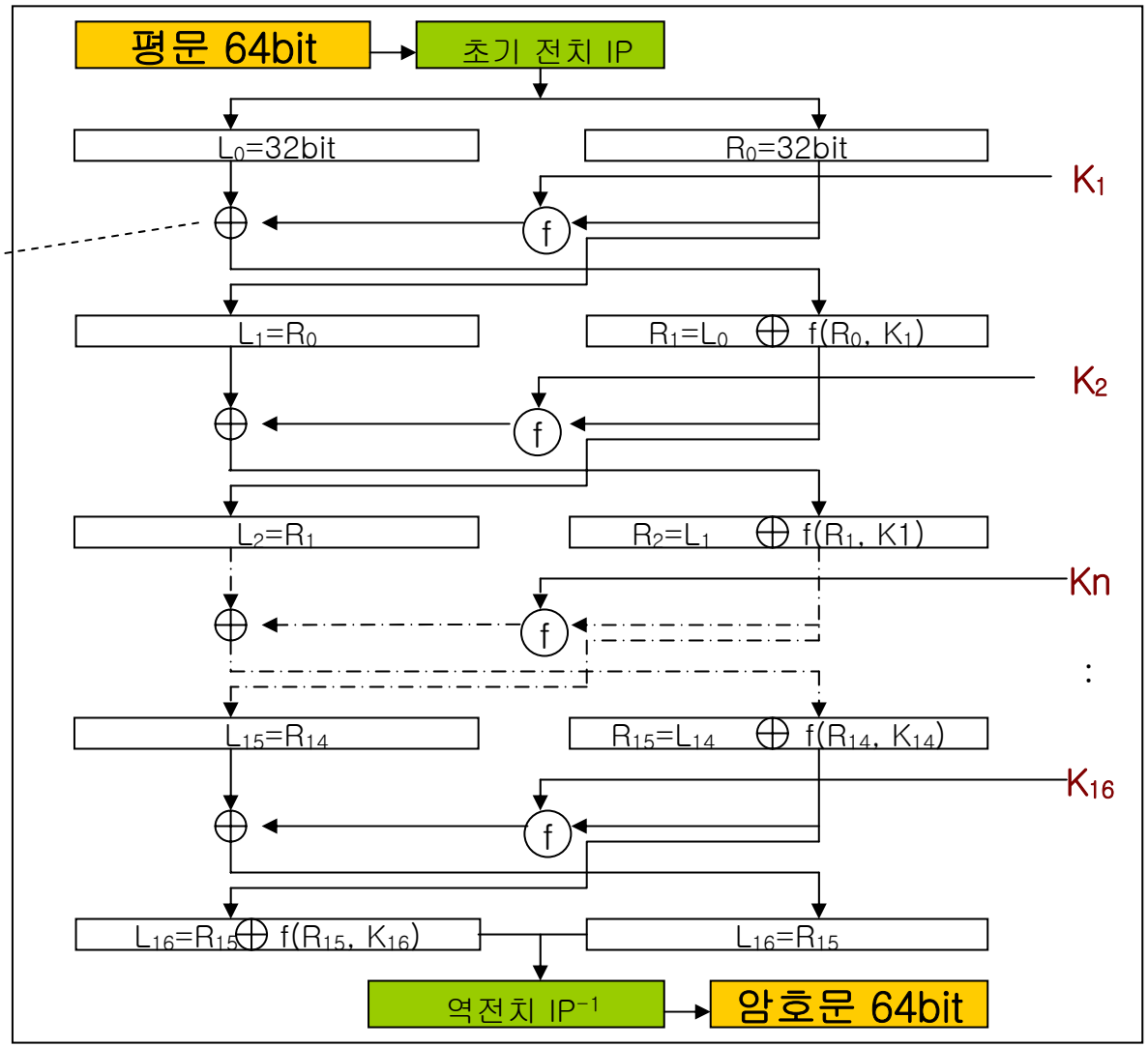
→약한 암호 알고리즘

대칭키 암호 알고리즘

◈ DES(Data Encryption Standard ; 데이터 암호화 표준)

- 치환 암호와 전치 암호를 교대로 반복 적용하여 강한 암호 알고리즘을 얻을 수 있다는 이론(C.E.Shannon)에 근거한 방식
- IBM이 개발하고 미국의 표준국(NIST)이 알고리즘 표준으로 채택
- 64bit 평문 블록(block)을 16번의 암호화를 통해 64bit블록의 암호문으로 변환 -64bit 씩 블럭화 과정을 거쳐 암호화
- 64 bit Block Key = 56bit 유효Key + 8bit Parity Check bit
  - 8비트당 1비트의 키는 검사용 비트로서, 키의 생성과 보관 과정에서 발생 할 수 있는 오류(error)를 파악
- DES 알고리즘의 기본동작
  - 전치 (종류: 평형 전치, 확대 전치, 축약 전치)
  - 치환 (S-Box에서 이루어짐)
  - mod 2 연산

대칭키 암호 알고리즘



배타적 논리합
(exclusive-OR)
f 알고리즘을 거친 키와
L_0에 대한 평문이 섞인다

XOR연산

A	B	output
0	0	0
0	1	1
1	0	1
1	1	0

16회 반복

[그림 6] DES의 원리

DES의 암호화 과정 – 3단계

1. 64비트 평문 M은 초기 전치 IP를 거쳐 M_0 생성, M_0 는 32비트씩 나누어져 L_0, R_0 로 나뉘어짐
   - 초기전치 IP는 평형전치로 [그림 7]과 같음
   - 58번째 비트를 1번째 비트로, 50번째 비트를 2번째 비트로 64비트의 위치를 변경시킴

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	26	8
57	49	41	33	25	17	9	1
59	51	43	35	2	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

[그림 7] 초기 전치
Initial permutation IP

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

[그림 8] 초기전치의 역전치
Inverse Initial permutation IP

DES의 암호화 과정 – 3단계 (cont.)

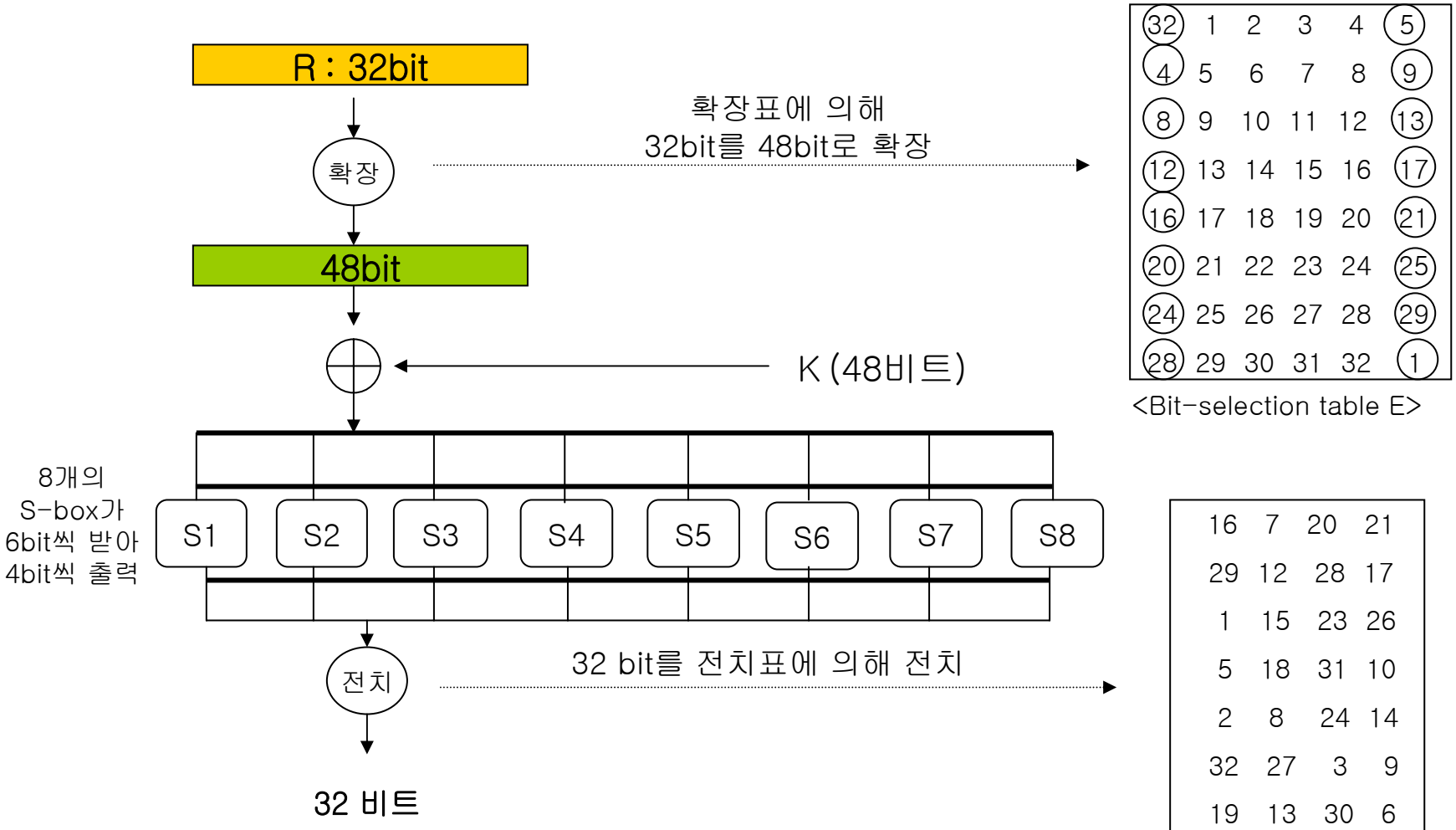
2. 초기 전치 출력 L_0, R_0 는 다음의 함수계산을 16회 반복함

- $L_i = R_{i-1}$, $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
  - $\oplus$ 는 mod 2 연산을 하는 EX-OR 를 의미
  - L_i, R_i 는 [그림 6]의 32비트씩의 중간 데이터
  - 서브키 K_i 는 48비트의 DES 암호화 키로써 $K_1, K_2, \ldots, K_{16}$ 의 값은 서로 다르며 16회 암호화 과정에 사용됨
  - 함수 f 는 S-Box를 포함한 치환 과정을 의미함

3. 은 초기 전치의 역전치인 IP^{-1}를 거쳐 64비트의 암호문이 됨 [그림 8]

- IP에서 58번째 비트가 1번째 비트로 전치되었기 때문에 반대로 IP^{-1}에서는 1번째 비트가 58번째 비트로 전치 되어야 함

대칭키 암호 알고리즘



32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

<Bit-selection table E>

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

<Permutation P>

[그림 9] *f* 알고리즘의 구조

대칭키 암호 알고리즘 – *f* 함수와 키 생성

- 함수 f 는 32비트의 R_i 입력과 48비트의 서브키 K_{i+1} 입력에 대하여 32비트의 중간 데이터로 치환하는 과정
  1. R_i 입력 32비트는 확대 전치 E를 거쳐 48비트로 확장됨
     - $E(R_i)$는 입력 R_i 의 32비트 중 16비트는 2회 나타나게 됨 → [그림 9]의 <Bit-selection table E>
  2. 확대 전치 출력 $E(R_i)$는 서브키 K_{i+1} 과 EX-OR된 후 6비트씩 8개로 나누어져 각각 8개의 S-Box에 입력이 됨
     - S*j*-Box의 입력 $B_j=b_1 b_2 b_3 b_4 b_5 b_6$ 은 S*j*-Box의표에서 $b_1 b_6$ 은 행을 지정하고 $b_2 b_3 b_4 b_5$ 는 열을 지정하여 행과 열이 만나는 지점의 숫자가 2진수로 바뀌어 S*j*-Box 출력이 됨 → S-box 환자표: 교재 61p.
  3. S-Box의 출력이 4비트 이므로 8개의 S-Box 출력의 합은 32비트가 됨
     - S-Box의 출력은 다시 평형 전치 P를 거쳐 f함수가 출력됨

  - f 함수의 출력은 R_i 와 mod 2 연산 후 L_{i+1} 이 됨
  - f 함수는 32비트의 R_i 입력과 48비트의 서브키 K_{i+1} 를 입력하여 전치와 S-Box의 치환으로 출력이 결정 됨

- f 함수 계산에 입력되는 서브키
  - 키 스케줄러에 입력되면 16개의 서브키가 출력됨

공개키 암호 알고리즘

▣ 공개키 암호 알고리즘

◈ 1976년 W.Diffie와 M.E.Hellman이 키의 안전한 분배를 위해 제시

◈ RSA방식이 대표적- 1978년 Rivest, Shamir, Adleman이 제시

◈ RSA 공개키 암호 알고리즘-소인수 분해 문제의 어려움을 이용

- 매우 큰 합성수 **n=pq(10^{150} 이상)** 일 때, n의 소인수 p와 q를 구하기 어렵다는 가정

◈키 생성 과정(A와B가 비밀통신을 할 경우)

① 수신자 B는 두개의 큰 소수 p와q를 생성하여 비밀키로 간직

② n=pq를 계산 (n 으로부터 p,q를 찾는 소인수 분해의 어려움을 이용)

③ (p-1)(q-1)=z를 계산, 이것과 서로 소가 되는 난수 e 선정

④ B는 n과 e값을 공개키로 제시(공개)

예) ① p=11, q=13으로 선정 → 이를 비밀키로 함

② n= pq=143

③ (p-1)(q-1)=120, 이와 서로 소가 되는 e를 11로 선정

④ B는 n=143과, e=11의 값을 공개키로 제시

공개키 암호 알고리즘

◈ RSA 공개키 암호방식 구현

·RSA 방정식

– **암호화: $c = x^e \bmod pq$ ($x^e - c = 0 \bmod pq$)**

·RSA 근의 공식

– 여기에서 유클리드 알고리즘을 사용하여 **$de+k(p-1)(q-1)=1$ 를 만족하는** (d, k)를 계산

– **복호화: $x=c^d \bmod pq$**

예) A가 B의 공개키 (n=143, e=11)를 이용하여 메시지 'K'(ASCII코드 값이 10진수로 75)를 암호화, B에게 전송하는 경우

·암호화 : A는 **$c = x^e \bmod pq$** 공식에 (n=143, e=11)를 대입, 만족하는 c의 값을 계산

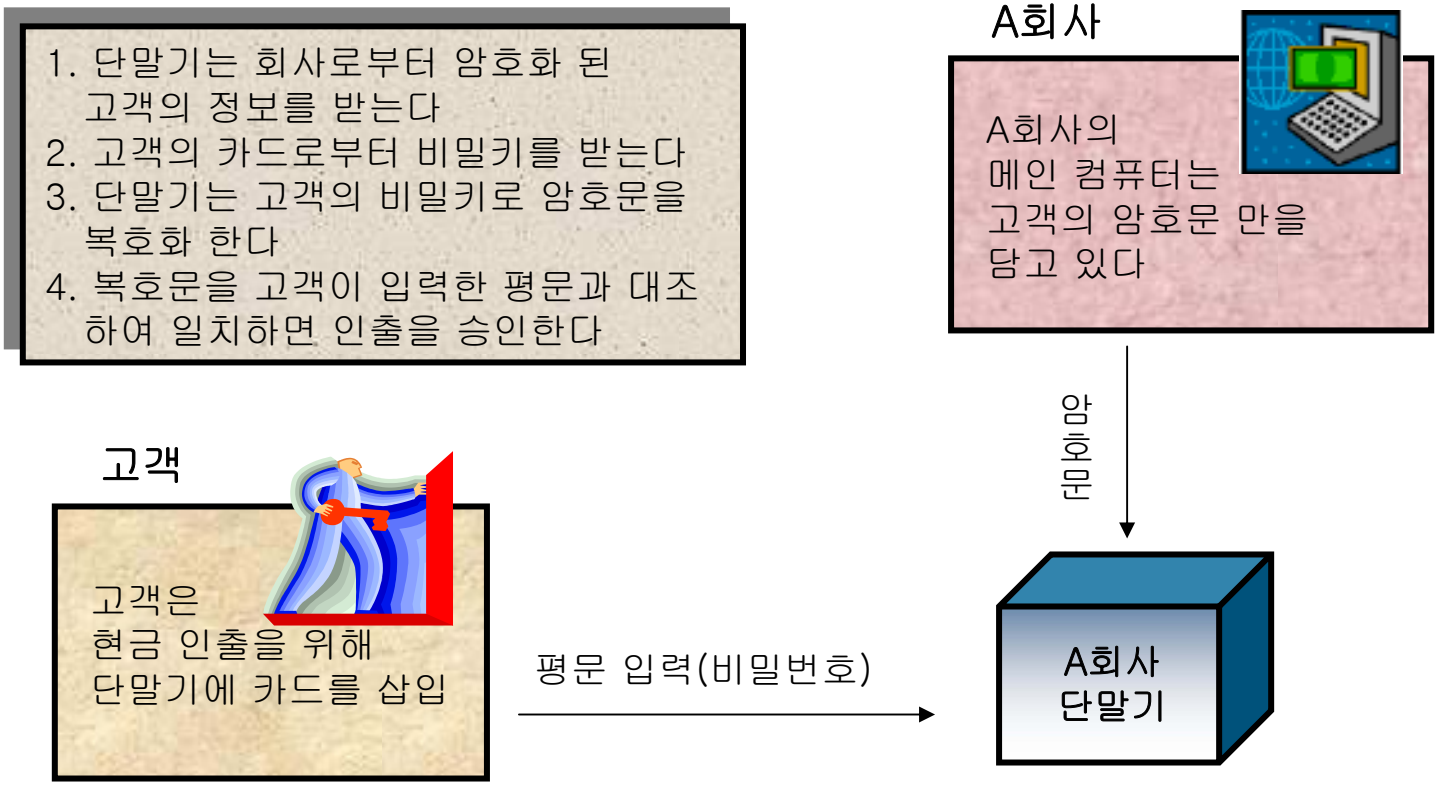
$c=75^{11} \bmod 143=108$ (이때 c는 메시지 '**K**'의 암호문)

·복호화 : B는 비밀키(p=11, q=13)를 가지고 근의 공식을 이용하여 복호화

① **$de+k(p-1)(q-1)=1$**를 만족하는 d와 k의 값을 계산 ($d=11, k=-1$)

② **$x=c^d \bmod pq$**에서 **$x=108^{11} \bmod 143=75=$'K'**

공개키 암호 알고리즘



[그림 10] 공개키 암호 알고리즘 적용 예

암호키 분배 프로토콜

▣ 암호키 분배 프로토콜

◈ Diffie와 Hellman이 이산대수가 어렵다는 가정 하에 설계

- **$Y=g^X \pmod p$**로부터 **$X=\log_g Y$** 를 구하기 어렵다는 가정

◈ 처리속도가 빠른 대칭키 방식으로 암·복호화 수행하고,
공개키 방식으로 공통의 암호화 키를 안전하게 전송

◈ 원리 : **$Y=g^X \pmod p$**→ 공개키, 둘만의 공통키는 다음과 같이 계산

· A와B는 사전에 200자릿수 크기를 갖는 소수p와 g의 값을 공개

① **$Y_A = g^{X_A} \pmod p$** : A는 자신만의 비밀키 X_A를 이용, Y_A값 생성→B에게 전송

② **$Y_B = g^{X_B} \pmod p$** : B는 자신만의 비밀키 X_B를 이용, Y_B값 생성→A에게 전송

③ **$K_{AB} = (Y_B)^{X_A} = g^{X_A X_B} \pmod p$** : A는 B로부터 받은 Y_B를 이용, 공통키로 사용

④ **$K_{AB} = (Y_A)^{X_B} = g^{X_A X_B} \pmod p$** : B는 A로부터 받은 Y_A를 이용, 공통키로 사용

* 주어진 공개 정보 (g, p, $Y_A = g^{X_A}$, $Y_B = g^{X_B}$)로부터 $g^{X_A X_B}$를 구하는 것이 계산상 불가능하다는 사실에 근거. 따라서 , 공통의 암호화 키는 A와 B만이 알 수 있음

전자서명

▣ 전자서명

◈ 전자적인 형태의 문서에 대한 서명

-날인의 전자적인 대체물

◈ 전자문서의 주인이 기대하는 사람과 맞는지 검증하기 위함

◈ 전자서명의 요건

- 유일성 : 정당한 서명자만이 전자서명을 생성
- 위조 불가능성 : 전자서명의 위·변조 불가능
- 진위 확인의 용이성 : 서명의 진위여부를 누구나 확인 가능
- 부인방지 : 서명자가 서명한 사실에 대한 부인을 방지

➔ 결국,전자서명 이라는 암호 프로토콜은 송신자 인증, 기밀성, 데이터 무결성이라는 목적을 달성해야 함

◈ 전자서명 방식

- 대칭키 암호방식 (생략)
- 공개키 암호방식을 주로 사용 (간략 설명)
- 효율성을 높이기 위해 일방향 해쉬함수를 함께 사용 (간략 설명l)

전자서명

◈ **공개키(RSA) 전자서명 방식**

– 송신자 A가 전자서명을 하는 경우

- 송신자 A의 비밀키➔ **p**(예:7), **q**(예:11)
- **de+k(p-1)(q-1)=1**를 만족하는 d(예 37)와 k를 계산
- 메시지 **m**(ASCII 코드값이 10진수로 18인 문자) 에 대한
  서명값 : **$x=m^d \bmod pq$**를 계산하여, 메시지 **m**과 서명 값 **x**를 검증자 B에게 전송함 (**$x=18^{37} \bmod 77=39$**)

– 수신자 B가 전자서명을 검증하는 경우

- 송신자 A의 공개키➔ **n=pq**(77), 임의의 난수 **e=13**
- 메시지 **m**과 서명값 **x**를 수신한 B는 "RSA 방정식"과 A의 공개키(n,e)를 이용하여 **$x^e - m = 0 \bmod n$** (예: $39^{13} - 18 = 0 \bmod 77$)를 만족하는지 검사 함

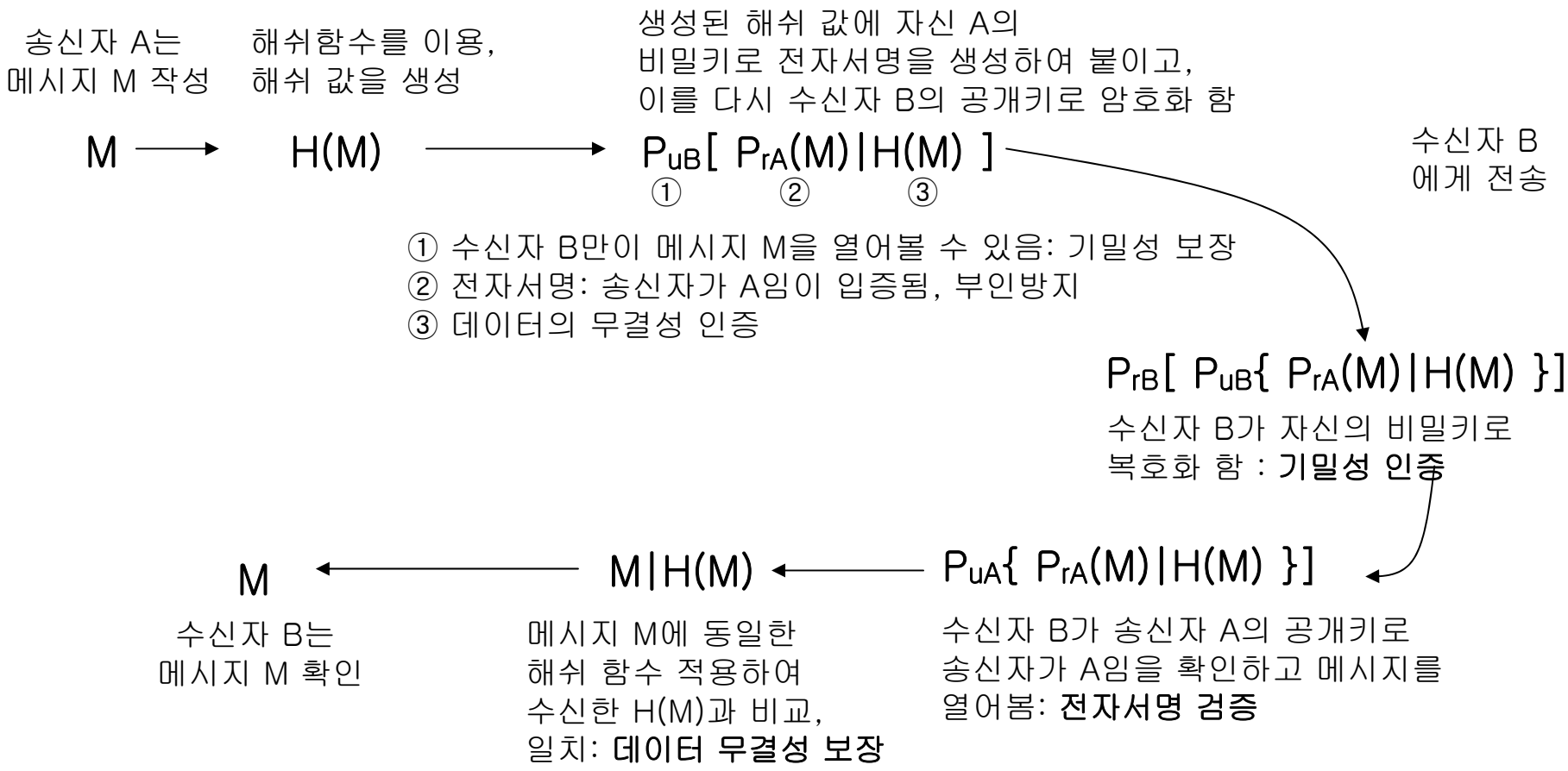
➔송신자 A: **$P_{rA}(M)$** 생성 후 수신자 B에게 전송

➔수신자 B: **$P_{uA}(P_{rA}(M))$** 로 서명 값을 확인하고 메시지를 받음

- Notation: P_{rA} : A의 비밀키, P_{uA} : A의 공개키, P_{rB} : B의 비밀키, P_{uB} : A의 공개키

전자서명

- 일방향 해쉬 함수를 사용한 전자서명 방식



[그림 11] 해쉬 함수를 사용한 전자서명 과정

전자서명

다중 전자서명 방식

◈ 동일한 전자문서에 여러 사람이 서명하는 것

-계약, 서명운동 등에 적용

◈ 단순 전자 서명방식의 단점을 보완

-단순서명을 반복 적용하는 번거로움을 방지

◈ 요건

- 서명문 길이의 고정 : 서명인의 수에 관계없이 길이가 일정
- 비밀 유지성 : 다중 서명 정보로부터 개인의 비밀정보 유추 불가
- 검증 가능성 : 정당한 참여자에 의해 서명 되었음을 누구나 입증 가능
- 공통성 : 각 서명자들은 모두 동일한 알고리즘을 이용하여 서명
- 부정 조기 검출성 : 생성중인 다중 서명을 중간 서명자들이 언제든 검증 가능

은닉 전자서명과 전자화폐

은닉 전자서명

- 전자상거래에서 사용자의 익명성을 보호하기 위해 등장
  - –신용카드: 사용자의 거래내역 추적 가능 → 개인 프라이버시 침해
  - –전자화폐: 은행과 상점의 결탁으로 사용자의 정보 노출 가능
- 프라이버시가 보장될 수 있는 전자화폐를 위하여 D.Chaum이 제안
- 메시지를 숨겨 제공자(provider:서명을 받는 사람)의 신원과 메시지를 연결시킬 수 없는 익명성 유지 가능
- 전자현금의 효율적인 사용이 가능하게 됨

전자화폐

- 디지털 형태의 전자현금
- 물리적인 화폐의 기능을 포함
- 발행, 지불, 결제단계로 구성

은닉 전자서명과 전자화폐

전자화폐 시스템

일반 RSA서명방식

1. 화폐 발행단계

① 은행은 공개키(n, e)와 비밀키(p, q, d)생성

② 은행은 전자문서 **m**을 준비

③ 전자문서 m에 대한 서명 $s=m^d \pmod p$ 생성

④ 은행은 서명 (m, s)를 사용자에게 전송

2. 화폐 지불단계

① 사용자가 서명(m, s)를 전자화폐로 사용

② 상점은 전자화폐의 정당성 확인 후 물품 제공

3. 결제단계

① 상점은 전자화폐(m, s)를 은행에 제시

② 은행은 (m, s)의 일련번호 기록

가
된

해당 은행의 DB에 접속, 화폐의
일련번호가 기사용 된 것인지 확인

은닉 전자서명과 전자화폐

◈ 은닉 **RSA** 전자서명 방식

1. 화폐 발행단계

① 은행은 공개키(n, e)와 비밀키(p, q, d)생성

② 사용자가 전자문서 m을 준비

– 자신만이 알고 있는 난수 r을 선택하여 z계산

$z=r^e \cdot m \pmod n$ → 은행에 전송

③ 은행은 비밀키 d를 이용하여 **z**에 대한 RSA의 서명

– **$s'=z^d=r \cdot m^d \pmod n$** 생성

④ 은행은 사용자에게 **s'** 전송 → 사용자의 계좌에서 해당 금액을 차감

⑤ 사용자는 r을 이용하여 전자화폐 생성

$-s=s'/r=m^d \pmod n$를 계산하면 (m,s)가 전자화폐가 됨

◈ 일반 RSA 서명방식과의 차이점

– 은행은 사용자가 준비한 m이 아닌, z에 대하여 서명 값을 생성하므로 m을 알 수 없음

– 나중에 m이 결제단계에서 은행의 DB에 기록이 되더라도 m이 누구의 것인지 알 수 없어, m으로부터 사용자의 정보 추적 불가

2. 화폐 지불단계, 3 결제 단계 : 일반 RSA 방식과 동일

전자투표

- 전자투표 시스템의 요구사항
  - 완전성(completeness)
    - 투표결과의 정확한 집계가 이루어져야 함
  - 비밀성(privacy)
    - 투표 내용과 투표자와의 관계는 비밀유지가 되어야 함
  - 재사용 불가(unreusability)
    - 투표자는 단 1회만 투표 가능(2중 투표 방지)
  - 공정성(fairness)
    - 투표 도중 집계결과가 나머지 투표에 영향을 주지 않아야 함
  - 자격성(eligibility)
    - 투표권이 없는 자의 투표 행위는 방지되어야 함
  - 검증성(verifiability)
    - 누구도 투표 결과를 위조할 수 없어야 함
  - 건전성(soundness)
    - 부정 투표자에 의한 선거 방해를 견뎌야 함

비밀분산

비밀 분산

- 중요 데이터를 몇 개의 조각으로 분산시켜 비밀유지
  - 한명에게 데이터를 전수했을 때의 유출 위험성을 방지
  - (m,n)-역치 방식(threshold scheme)이 제안됨
- **(m, n)역치 방식의 특성**
  - ① n개의 비밀 조각 중 m개 이상이 있으면 비밀 재구성 가능
  - ② m-1개 미만의 조각으론 비밀 재구성 불가능
- m-1차 다항식과 Lagrange 공식을 이용한 역치 방식
  - T가 n명의 사용자에게 비밀 s를 분산시킨다고 가정
  - ① T는 비밀정보 s를 상수항으로 하고 m-1차 다항식 f를 임의 선택

→ $f(x)= s + a_1x + a_2x^2 +...+ a_{m-1} x^{m-1} (\mathrm{mod}\ p)$

($a_1, a_2, \ldots, a_{m-1}$ 은 임의의 정수, p는 $a_1, a_2, \ldots, a_{m-1}$ 보다 큰 임의의 소수)

② 각각의 분산 유지자 i(1≤i≤n)에게 비밀 조각 $s_i=f(i)$를 계산하여 전송

③ n개 중 m개를 가지고 비밀 s를 재구성

→ 비밀 조각 s_i (1≤i≤m) 를 입력 값으로 다음의 Lagrange 공식 이용 (910p.)

비밀분산

▣ 비밀분산의 예

◈ ASCII 코드 값이 10진수로 11인 비밀 s를 (3, 5)-역치 방식으로 5명의 사용자에게 분배할 경우 (분배자는 p=13, a=7, b=8, s=11 임의 선정)

* $f(x)=ax^2 + bx + s \pmod p$ → $f(x)=7x^2 + 8x + 11 \pmod{13}$

$s_1 = f(1) = 7 + 8 + 11 = 0 \pmod{13}$

$s_2 = f(2) = 28 + 16 + 11 = 3 \pmod{13}$

$s_3 = f(3) = 63 + 24 + 11 = 7 \pmod{13}$

$s_4 = f(4) = 112 + 32 + 11 = 12 \pmod{13}$

$s_5 = f(5) = 175 + 40 + 11 = 5 \pmod{13}$

* s_2, s_3, s_5 로 비밀을 재구성할 경우→사용자들은 a, b를 구하여 s완성

→ $f(x)=ax^2 + bx + s \pmod p$에 각각의 X값을 대입

$f(2) = a \cdot 2^2 + b \cdot 2 + s = 3 \pmod{13}$

$f(3) = a \cdot 3^2 + b \cdot 3 + s = 7 \pmod{13}$

$f(5) = a \cdot 5^2 + b \cdot 5 + s = 5 \pmod{13}$

➔ 방정식의 해는 a=7, b=8, 비밀 s=11

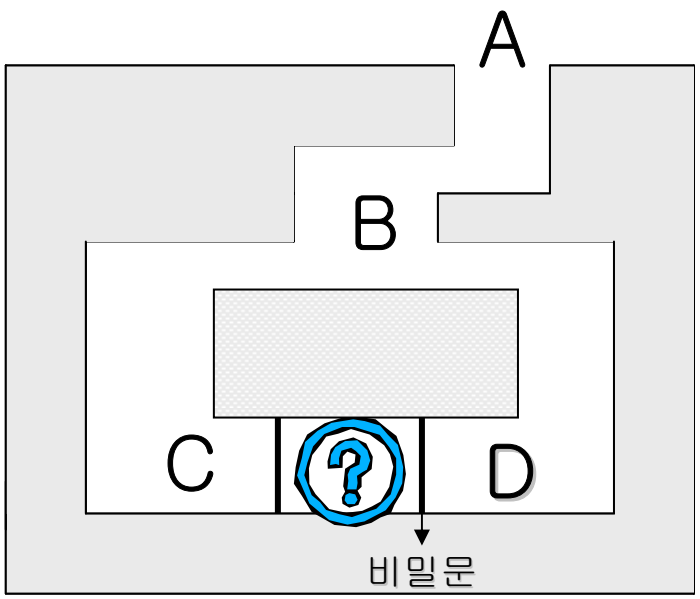
영지식 증명 프로토콜

■ 영지식 증명 프로토콜(zero-knowledge proof)

◈증명자(P)가 어떤 사실의 정당성에 관한 사실만을 검증자(V)에게 전달하는 방식

– 비밀의 내용을 드러내지 않으면서 비밀을 갖고 있다는 것을 확신 시키는 방식

◈ 원리 : P(proover)가 V(verifier)에게 증명할 경우



[그림 13] 영지식 동굴

① V가 A지점에 서 있는다
② P가 C또는 D지점으로 간다
③ V가 B지점으로 간다
④ V는 P에게 다음 중 하나를 외친다
  a.좌측 통로로 나오시오
  b.우측 통로로 나오시오
⑤ P와 V는 위의 과정을 n회 반복한다

* P가 비밀 정보를 가지고 있지 않을 경우에 V의 요구에 응할 확률 →1/2
* n번 반복한 후에 P가 비밀 정보를 알고 있음에 대한 V의 확신률 →$1-(1/2)^n$

영지식 증명 프로토콜

◈ 영지식 증명 시스템의 특성

- P가 V에게 어떤 사실의 정당성을 증명한 후에, V는 제3자에게 그 사실을 다시 증명 할 수 없다

  ▤ 녹화된 대화 내용(31p. ①~⑤)을 보여줄 경우라도 P와 V가 모의해서 4단계의 질문을 사전에 정해놓으면, P가 비밀정보를 모르더라도 같은 상황을 연출가능

- 비밀정보를 모르는 제3자라도 P와 V의 대화내용을 흉내낼 수 있다. 즉, V는 P가 비밀정보를 갖고 있다는 사실 외에는 어떤 정보도 알 수 없음

참고문헌

- 김승주, 박성준, 이임영, 원동호, “암호알고리듬과 암호프로토콜” Telecommunications Review,제10권5호,pp.901-914, 2000.10.
- 원동호, 현대 암호학, 그린출판, 2003
- Henk C.A. van Tilborg, An Introduction to Cryptology, Kluwer Academic Publishers, 1988.