7주차: 암호의 이해

ChulSoo Park

School of Computer Engineering & Information Technology

Korea National University of Transportation

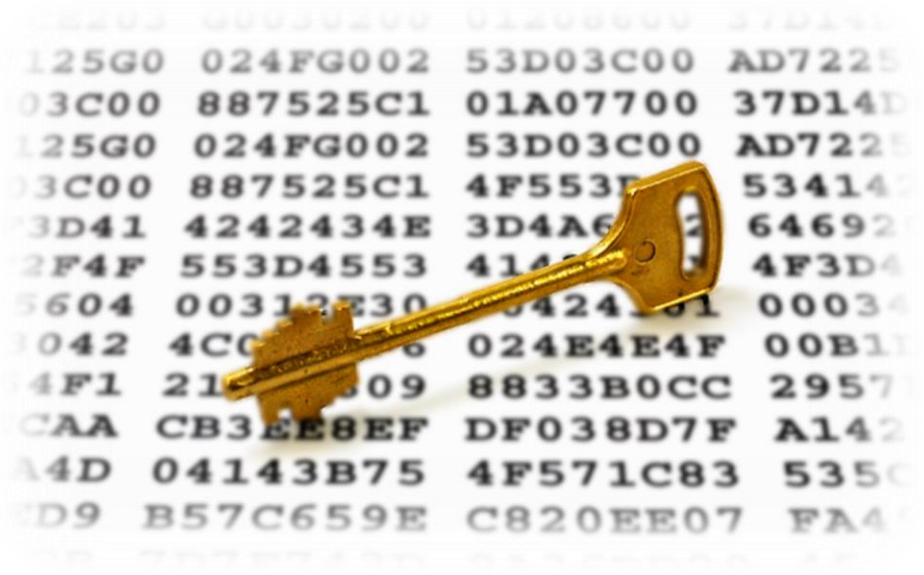
E-Mail : pcs8321@naver.com

학습목표 (7주차)

- 암호 원리의 이해
- 대칭 암호의 원리와 기능 이해
- 비대칭 암호의 원리와 기능 이해
- 해시 알고리즘의 원리 이해

07 CHAPTER

암호의 이해



CONTENTS

1. 암호의 개념과 원리
2. 대칭 암호화 방식
3. 비대칭 암호화 방식
4. 해시

1. 암호의 개념과 원리

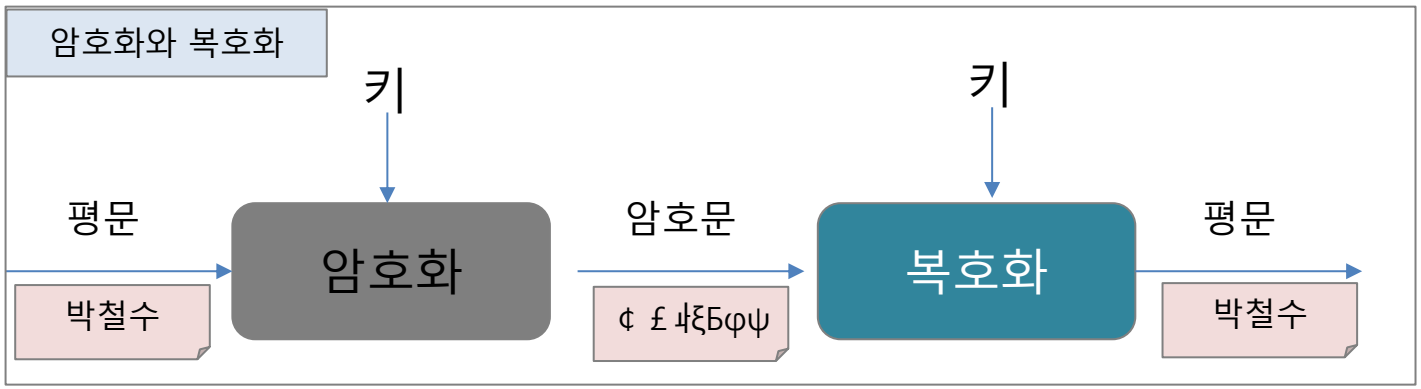
암호화의 필요성

- **기밀성(Confidentiality)** : 데이터가 의도하지 않은 자에게 노출되지 않아야 함.
- **무결성(Integrity)**: 네트워크를 통해 송수신되는 데이터가 임의로 조작되거나 삭제되지 않아야 함.
- **인증(Authentication)** : 당사자가 서로의 신원을 확실하게 검증할 수 있어야 함.
- **부인방지(Non-repudiation)** : 데이터를 보낸 사람이 보낸 사실을 부인하거나, 받은 사람이 받은 사실을 부인할 수 없어야 함.

1. 암호의 개념과 원리

암호문

- 비밀을 유지하기 위해 당사자만 알 수 있도록 꾸민 약속 기호. (반대 : 평문)
- 최초의 암호는 기원전 480년 페르시아에서 살던 데마라토스가 페르시아의 침략 계획을 나무판에 조각하고, 밀랍을 발라 스파르타에 보낸 것
- 암호는 암호문이 노출되더라도 정보를 숨길 수 있어야 함
  - ✓ 암호화: 평문을 암호문으로 바꾸는 것
  - ✓ 복호화: 암호문을 평문으로 바꾸는 것
  - ✓ 암호화 알고리즘: 암호화나 복호화를 수행할 때 양쪽이 알고 있어야 할 수단
  - ✓ 암호화 키: 약속한 규칙



1. 암호의 개념과 원리

- 암호화 방식 - 전치법
  - ✓ 단순히 메시지에 들어 있는 문자 위치를 바꾸는 방법
  - ✓ 미리 정해둔 문자 배열 규칙으로 암호화와 복호화 수행
  - ✓ 스파르타에서 군사용으로 사용하던 봉 암호화도 전치법의 일종 (키테일 암호화)
  - ✓ 예) 박철수김영희 → 철박김수희영

- 암호화 방식-대체법
  - ✓ 메시지의 글자를 다른 글자로 대체하여 암호화하는 방법
  - ✓ 적절한 배합을 찾으면 쉽게 복호화할 수 있는 전치법의 문제를 해결하기 위해 등장
  - ✓ **단일 치환**과 **다중 치환**으로 나눌 수 있음

- 알파벳 한 글자를 다른 하나의 글자로 대체하는 방식(A → C)
- 시저 암호화, 모노 알파베틱 암호화가 있음

- **시저 암호화**
  - ✓ 알파벳 스물여섯 자를 세 자 또는 네 자씩 오른쪽으로 이동한 뒤 해당되는 글자로 변환하여 암호화하는 것
  - ✓ 500년 동안이나 사용된 방법이지만, 암호화가 가능한 경우의 수가 26에 불과한 매우 취약한 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
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

EHFDUHIXO IRU DVVDVVLQDWRU

→ BE CAREFUL FOR ASSASSINATOR

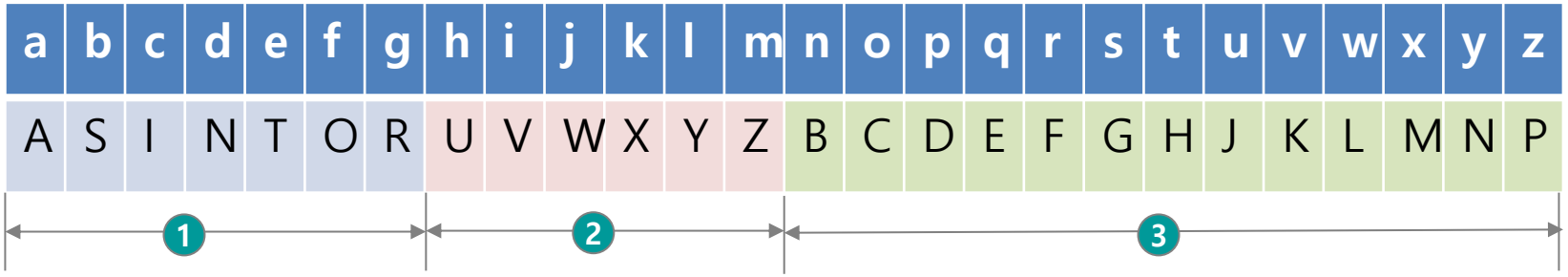
모노 알파베틱 암호화

- 알파벳 스물여섯 자를 각각 다른 알파벳에 일대일 대응시켜 알파벳을 암호화하는 것
- 모노 알파베틱 암호문을 복호화하려면 알파벳 대칭표가 있어야 함
- 모노 알파베틱으로 암호화한 함호문은 26!
- 예) ASSASSINATOR 대칭표 만들기

  ① 중복 제거 ASINATOR

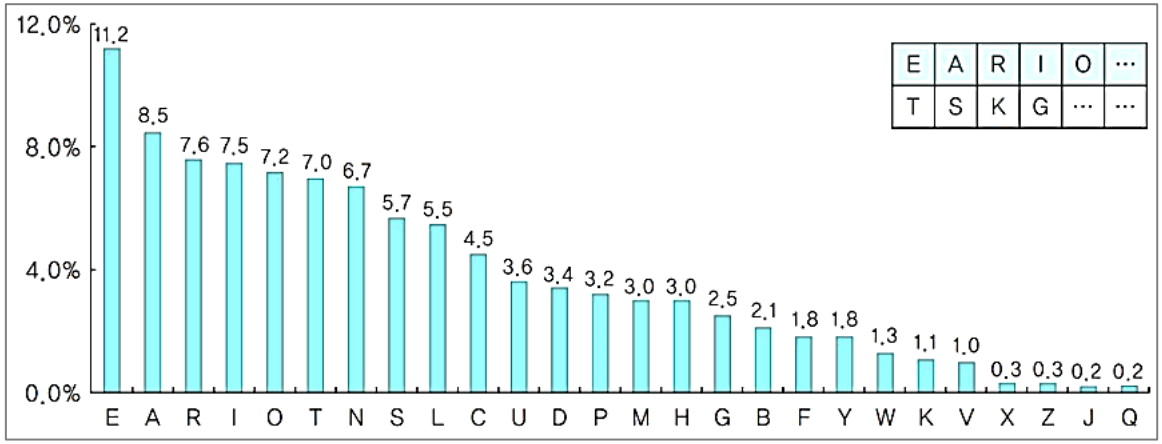
  ② R ~ Z까지 중 앞에 나온 알파벳을 제외하고 뒤에 적음

  ③ 알파벳 26자 중 나오지 않은 알파벳 적음



단일 치환 암호화의 특징

- 단일 치환 암호법은 키워드를 몰라도 복호화가 가능
- 빈도 분석법: 알파벳 스물여섯 자가 문장에서 비슷한 빈도로 사용된다는 통계에서 착안한 것
- 단일 치환 암호법의 암호문에 사용된 알파벳의 빈도를 계산해서 통계와 비교해 알파벳을 확인할 수 있음.(peopre → people로 보정)



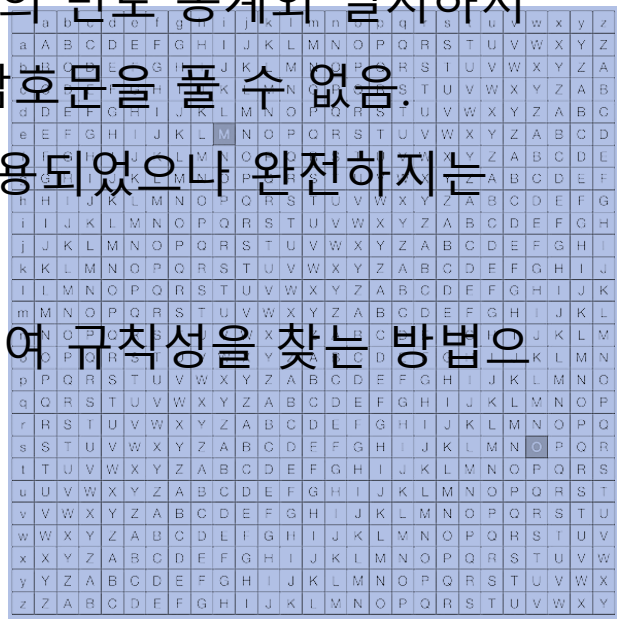
(옥스포드 영어사전(9판)의 알파벳 빈도

다중 치환(Polyalphabetic Substitution) 암호화의 원리

- 암호화 키와의 매핑에 따라 알파벳 하나가 여러 가지 다른 알파벳으로 대체되어 암호화되는 것.
- 예) 암호문에 a가 경우에 따라서 c가 될 수 있고 r이나 t가 될 수도 있음.
- 다중 치환 암호화 방식
  - ✓ 비즈네르((Vigenere) 암호화
  - ✓ 플레이페어(Playfair) 암호화
  - ✓ 힐(Hill) 암호화

비즈네르 암호화

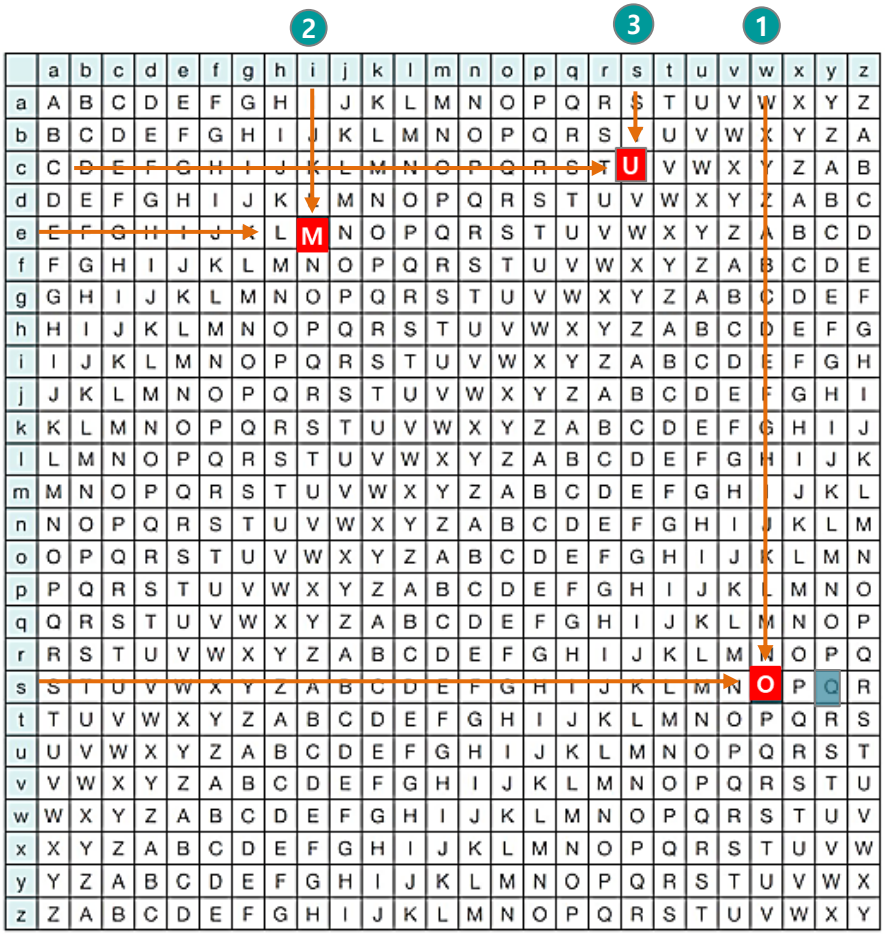
- 26×26의 알파벳 대칭표를 이용하여 암호화하고자 하는 평문과 암호화 키를 매핑하고 암호화와 복호화를 수행하는 방식.
- 16세기에 프랑스 외교관 블레즈 비즈네르가 만듬.
- 암호문에 사용된 문자의 빈도가 일반적인 문자의 빈도 통계와 일치하지 않아 앞서 살펴본 단순한 빈도 분석법으로는 암호문을 풀 수 없음.
- 비즈네르 암호화 방식은 17~18세기에 널리 사용되었으나 완전하지는 않았음
- 19세기에 찰스 배비지는 빈도 분석법을 이용하여 규칙성을 찾는 방법으로 비즈네르 암호를 복호화.



비즈네르 암호화

암호화 과정

- 암호화하려는 평문: 'wish to be free from myself'
- 암호화 키: 'secret is beautiful'
- ① 평문의 첫 문자인 w를 비즈네르 표의 가로축에서 찾고 암호화 키의 첫 문자인 s를 세로축 에서 찾으면 O에 대칭
- ② 평문의 두 번째 문자 i와 암호화 키의 두 번째 문자 e를 비즈네르 표에서 찾으면 M에 대칭
- ③ 평문 s(가로), 암호키 c(세로)의 대칭 값 U
- 평문 'wish to be free from myself'는 'OMUY XH JW GVEY YZTG XQWGCJ' 라는 암호문



비즈네르 암호화 예

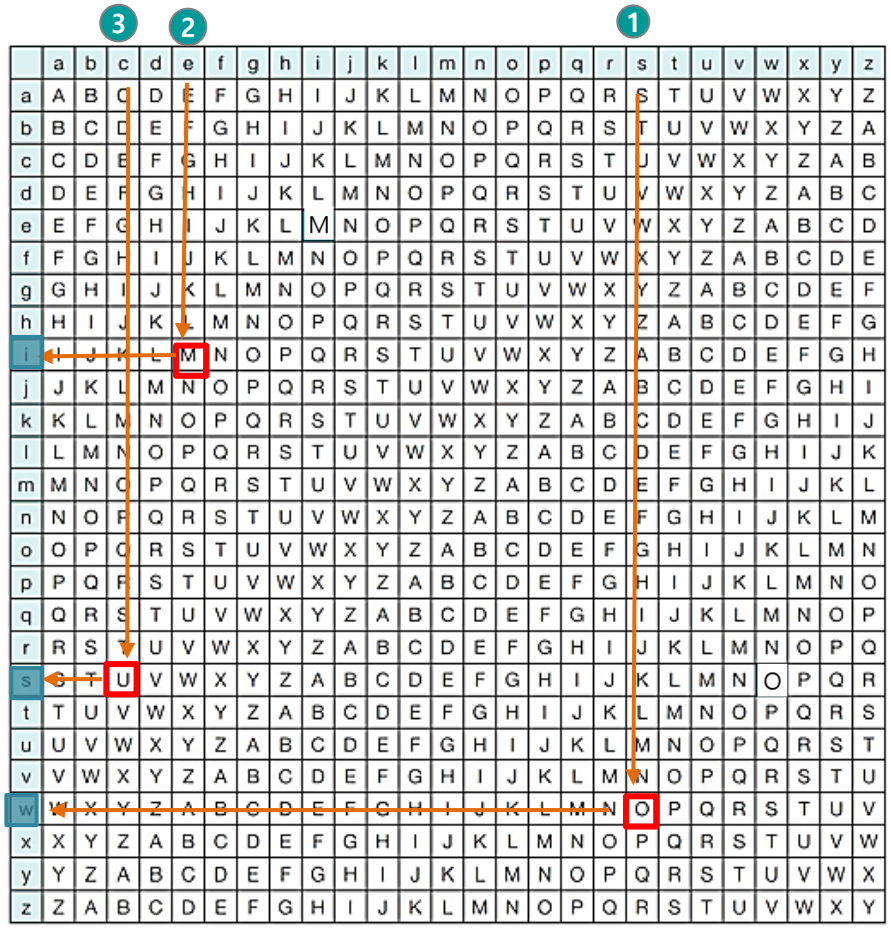
평문	W	i	s	h	t	o	b	e	f	r	e	e	f	r	o	m	m	y	s	e	l	f
암호회 키	s	e	c	r	e	t	i	s	b	e	a	u	t	i	f	u	l	s	e	c	r	e
	O	M	U	Y	X	H	J	W	G	V	E	Y	Y	Z	T	G	X	Q	W	G	C	J

평문의 e는 W,E,Y,G로 치환

비즈네르 암호화

비즈네르 복호화 과정

- ① 암호화 키의 첫 문자인 s를 비즈네르표의 가로축에서 찾고
- 암호문의 첫 문자인 O를 밑으로 내려가면서 찾아 세로축 값(w)을 찾음
- ② 암호키 e, 암호문 M → 세로축 값(i)
- ③ 암호키 c, 암호문 U → 세로축 값(s)



암호회 키	s	e	c	r	e	t	i	s	b	e	a	u	t	i	f	u	l	s	e	c	r	e
암호문	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	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	f

플레이페어 암호화

- 1854년 찰스 휘트스톤이 개발했고 초기에는 어렵다는 이유로 사용되지 않음
- 라이언 플레이페어를 통해 널리 알려지면서 제1차 세계 대전 때 영국 육군이 야전 표준 시스템으로 사용.
- 제2차 세계대전 때는 미국 육군을 비롯한 연합군이 사용.
- 2개로 이루어진 문자 쌍을 다른 문자 쌍으로 대체하는 암호화 방법.
- 보통 정사각형 암호판(5X5) 안에 영어 알파벳을 배열한 것으로 대체하는 방식.
- 5 X 5 암호화 테이블에 26알파벳을 나열하며 한 칸이 부족 I와J or Q와 J 같은 칸
- NIGHTTIME 중복 제거 → NIGHTME

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

플레이페어 암호화 테이블

플레이페어 암호화

- 암호화 키 ASSASSINATOR에서 중복 제거 → ASINTOR로 플레이페어 암호 테이블 만들기.

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

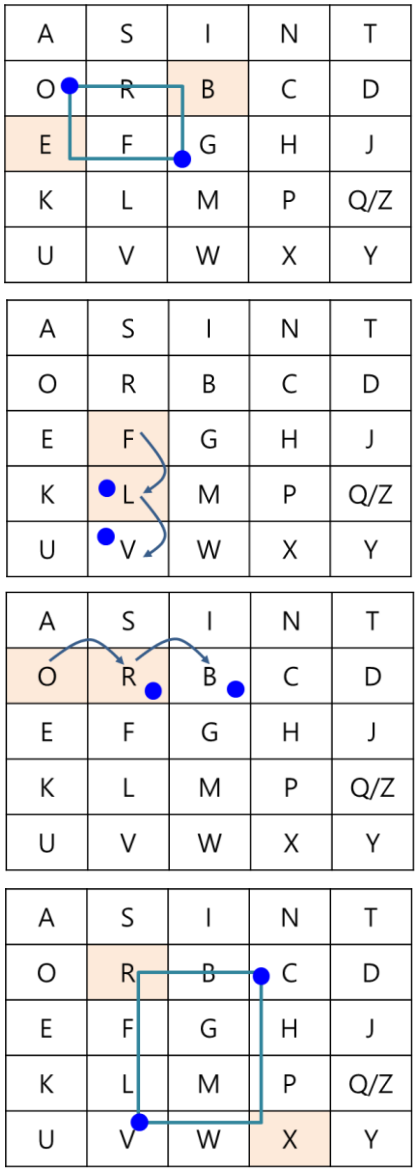
- 주어진 문장으로 평문을 2개의 문자 쌍으로 만듬.
- BE CAREFUL FOR ASSASSINATOR (문자가 같거나 마지막에 하나이면 '**X**' 추가

BE	CA	RE	FU	LF	OR	AS	SA	S**X**	SI	NA	TO	R**X**

플레이페어 암호문 만들기

평문	BE	CA	RE	FU	LF	OR	AS	SA	SX	SI	NA	TO	RX
	OG	ON			VL	RB							CV

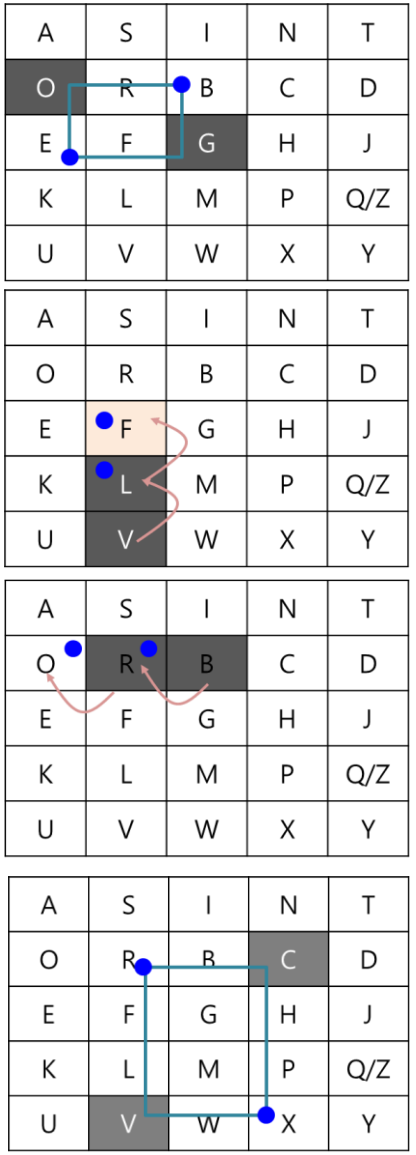
- 암호화 하려는 두문자 (BE) 암호화 테이블의 B,E의 행과 열이 만나는 위치의 O,G로 대체
- FL과 같이 같은 열에 있으면 L → V, F → L이 됨.
- OR과 같이 같은 행에 있으면 O→ R, R→B가 됨.
- RX는 C,V가 됨



플레이페어 복호화

암호문	OG	ON	OF	EV	VL	RB	SI	IS	NV	IN	TS	AD	CV
	BE	CA			LF	OR							RX

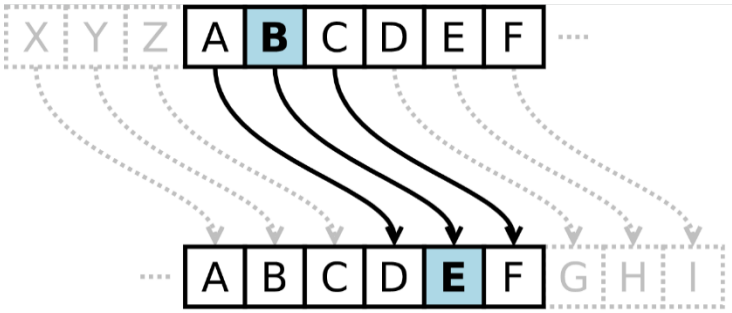
- 암호문 OG → BE
- 암호문 VL → LF
- 암호문 RB → OR
- 암호문 CV → RX



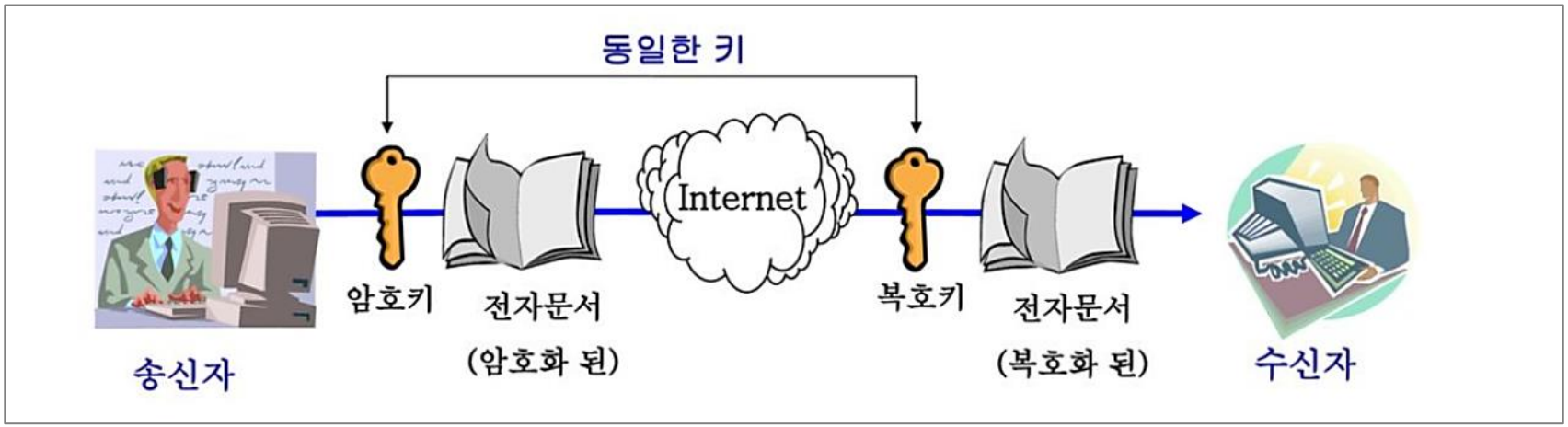
2. 대칭 암호화 방식

고전적인 암호화는 강도가 낮음

- 고전적 암호화는 비교적 쉽게 복호화 가능.
- 강력한 암호화 알고리즘은 **혼돈(Confusion)**과 **확산(Diffusion)**의 특성을 이용함.
- **혼돈** : 암호문의 통계적 성질과 평문의 통계적 성질의 관계를 어렵게 하여 만드는 것
- **확산** : 각각의 평문 비트와 키 비트가 암호문의 모든 비트에 영향을 주는 성질을 이용하는 것.
- 현대의 대칭 암호화 알고리즘은 혼돈과 확산의 기능을 대포 강화한 것.



2. 대칭 암호화 방식

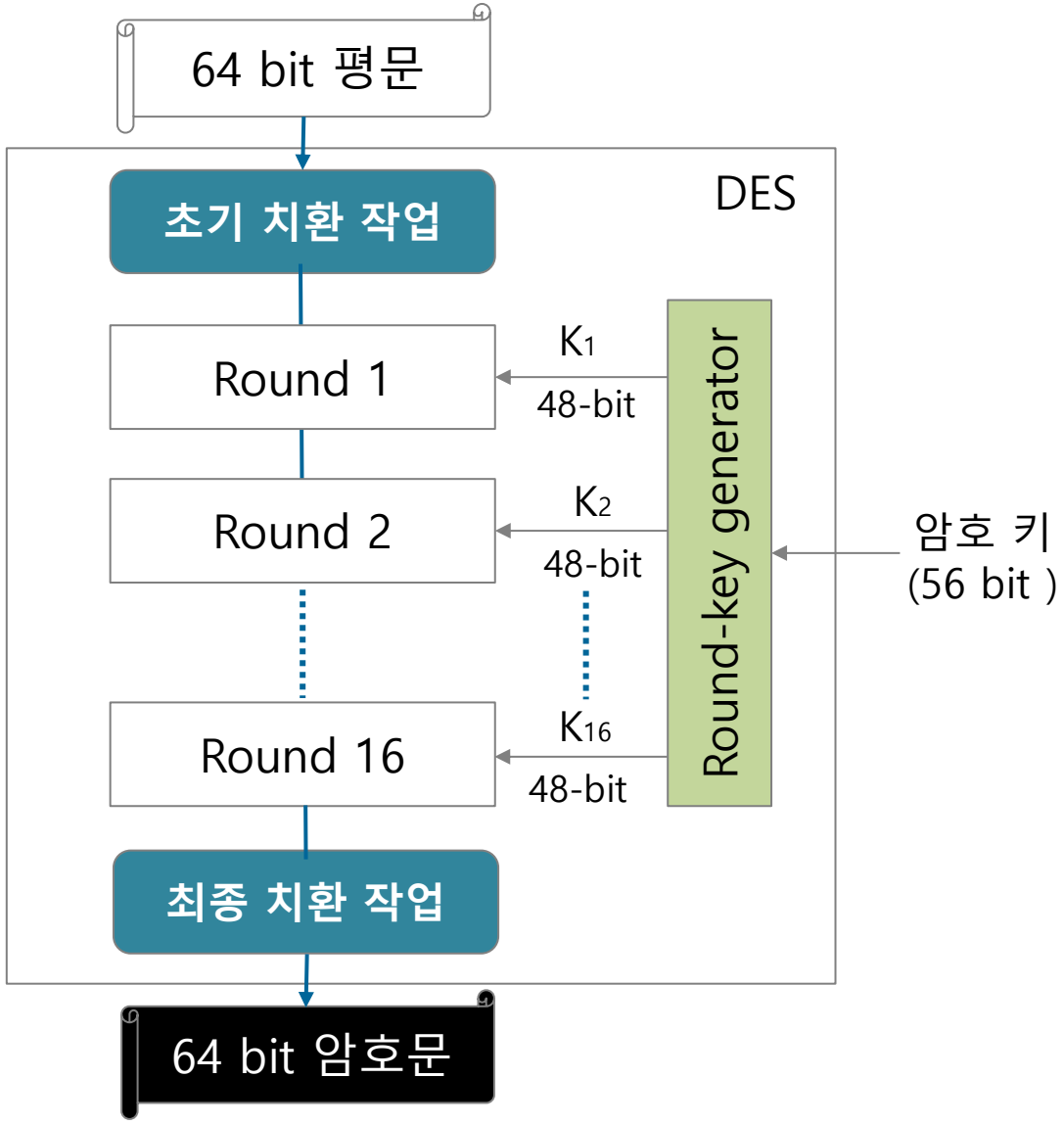


- **DES**는 대칭형 암호화(비밀 키 암화화)이다. 즉 암호화와 복호화 키가 동일
- 비대칭에 비하여 암호화 속도가 빠름
- 1972년 미국 상무부의 **NBS에서 정보 보호를 목적으로 공모한 암호화 알고리즘**
- IBM의 바터 투흐만과 칼 마이어가 개발
- 1977년 1월 국립표준기술연구원(**NIST,** (National Institute of Standards and Technology)**에 의해 암호화 표준으로 결정**
- DES는 64비트의 블록 암호화 알고리즘으로 56비트 크기의 암호화 키로 암호화

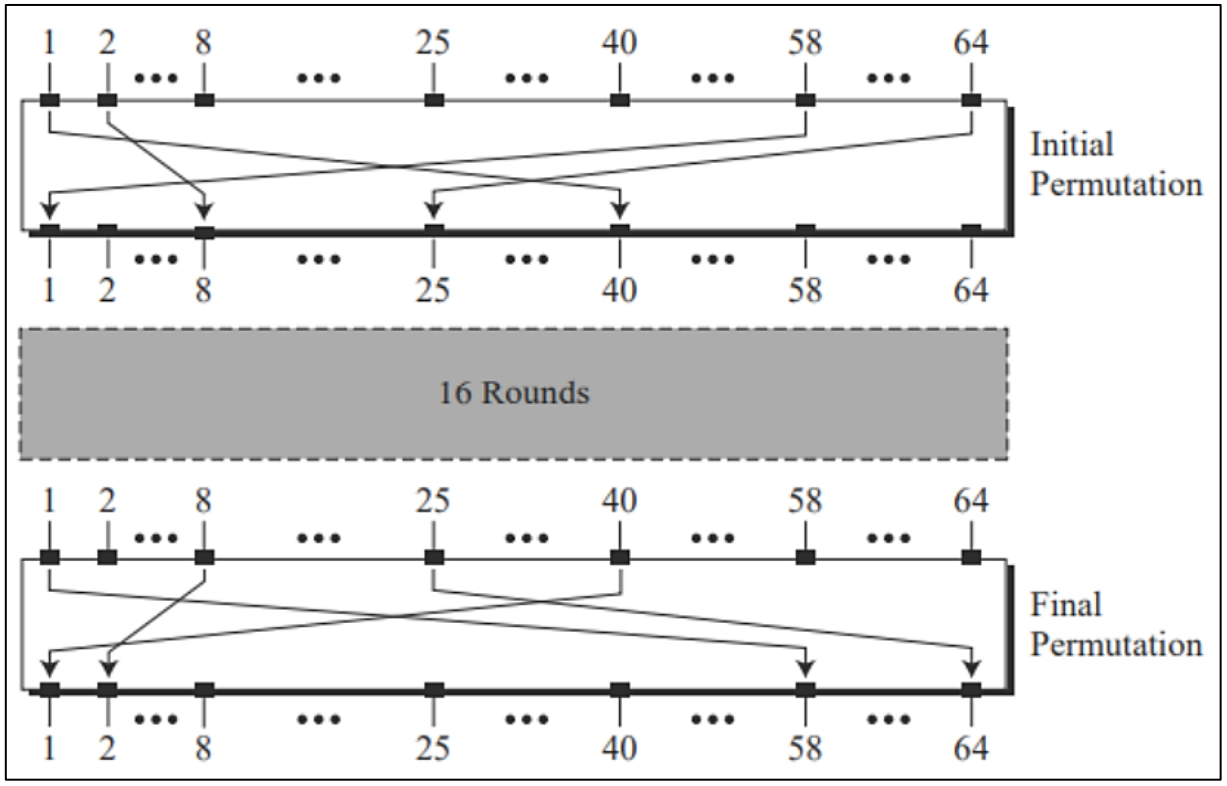
DES 암호화 과정

DES 암호화 과정에는 크게 3가지로 나눌 수 있다.

① Initial Permutation & Final Permutation
② Round Function
③ Round-key generator

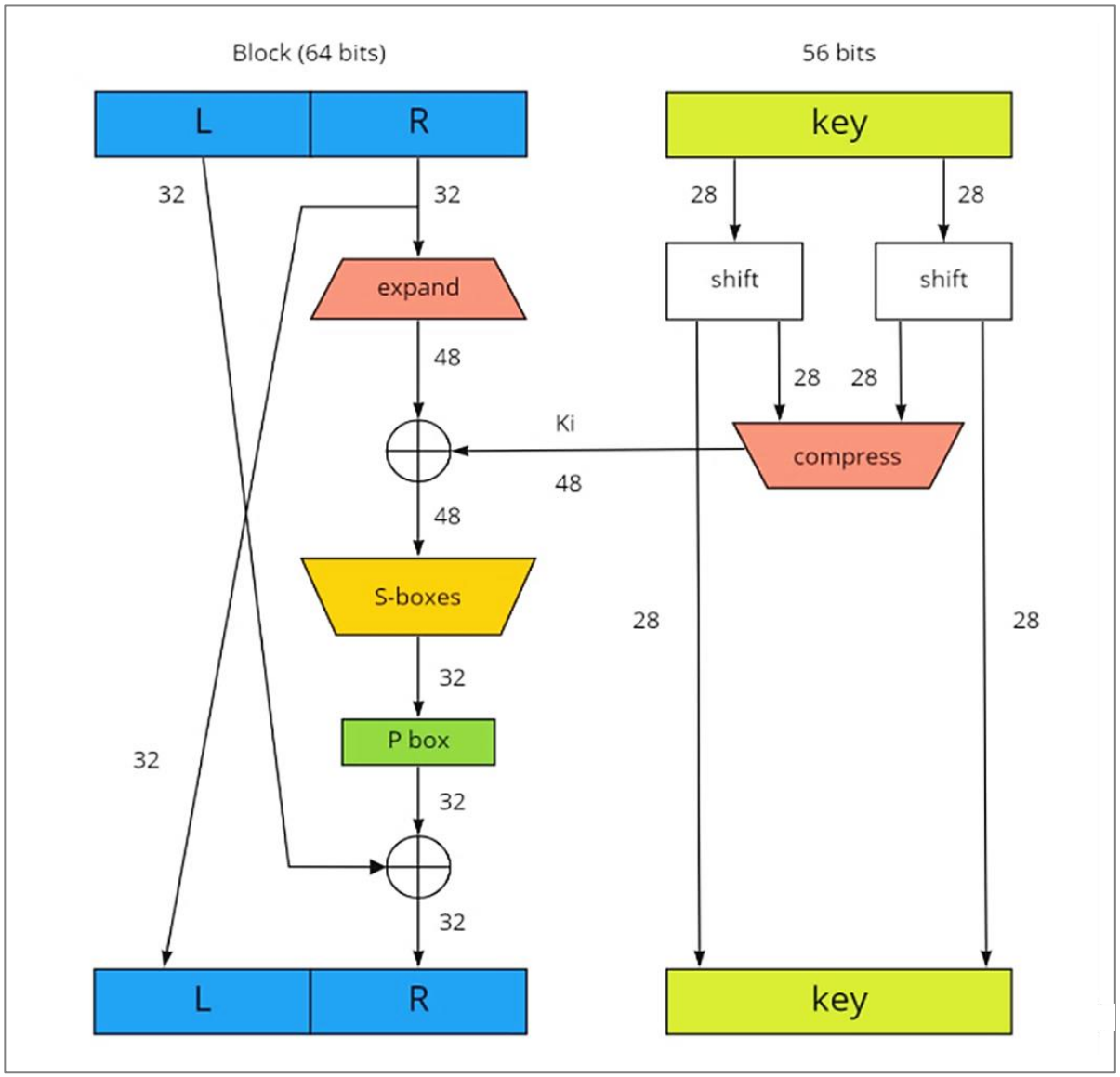


① Initial Permutation & Final Permutation



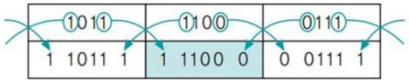
Initial Permutation	*Final Permutation*
58 50 42 34 26 18 10 02	40 08 48 16 56 24 64 32
60 52 44 36 28 20 12 04	39 07 47 15 55 23 63 31
62 54 46 38 30 22 14 06	38 06 46 14 54 22 62 30
64 56 48 40 32 24 16 08	37 05 45 13 53 21 61 29
57 49 41 33 25 17 09 01	36 04 44 12 52 20 60 28
59 51 43 35 27 19 11 03	35 03 43 11 51 19 59 27
61 53 45 37 29 21 13 05	34 02 42 10 50 18 58 26
63 55 47 39 31 23 15 07	33 01 41 09 49 17 57 25

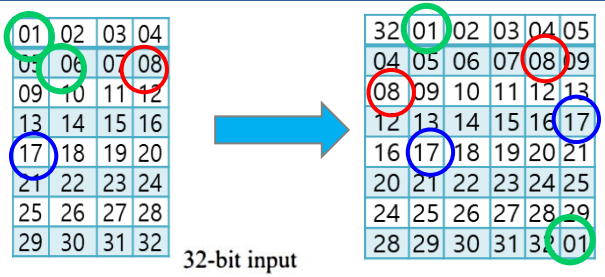
② Round Function

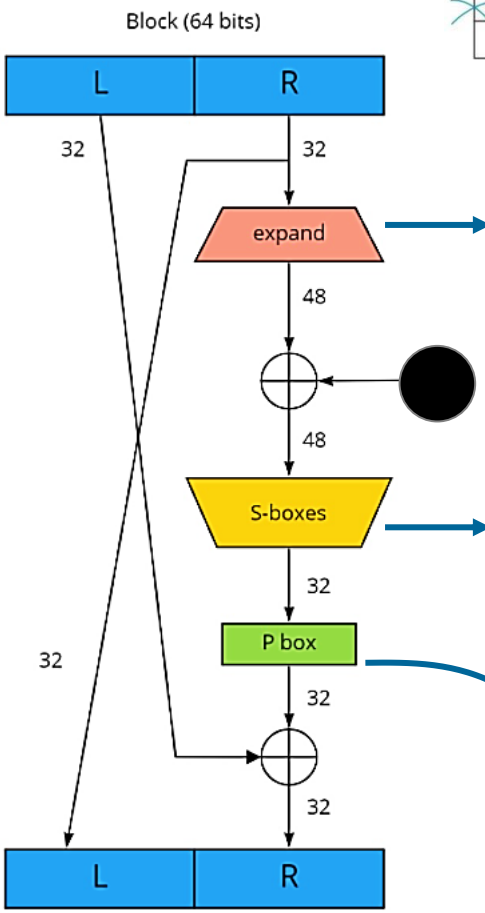


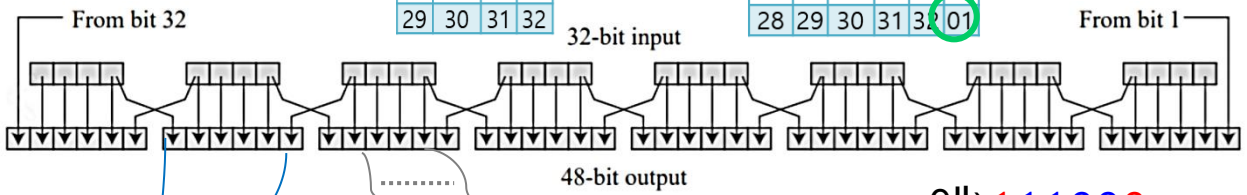
② Round Function

그림 7-14









Input bit (1,2,3,4)

예)111000

Input bit (0,5)

	00 00	00 01	00 10	00 11	01 00	01 01	01 10	01 11	10 00	10 01	10 10	10 11	11 00	11 01	11 10	11 11
0 0	11 10	01 00	11 01	00 01	00 10	11 11	10 11	10 00	00 11	10 10	01 10	11 00	01 01	10 01	00 00	01 11
0 1	00 00	11 11	01 11	01 00	11 10	00 10	11 01	00 01	10 10	01 10	11 00	10 11	10 01	01 01	00 11	10 00
1 0	01 00	11 01	11 10	10 00	11 01	00 10	10 11	11 11	11 00	10 01	01 11	00 11	10 10	01 01	00 00	00 00
1 1	11 11	11 00	10 00	00 10	01 00	10 01	00 01	01 11	01 11	10 11	00 11	11 10	10 10	00 00	01 10	11 01

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

15	6	19	20	28	11	27	16	0	14	22	25	4	17	30	9
1	7	23	13	31	26	2	8	18	12	29	5	21	10	3	24

③ Round-key generator

56 bits (0,1,2,...,55) key

Left half key bits: LK (28bits)

49	42	35	28	21	14	7
0	50	43	36	29	22	15
8	1	51	44	37	30	23
16	9	2	52	45	38	31

Right half key bits: RK (28bits)

55	48	41	34	27	20	13
6	54	47	40	33	26	19
12	5	53	46	39	32	25
18	11	4	24	17	10	3

For rounds i=1,2,...,16

Let LK = (LK circular shift left by r_i)

Let RK = (RK circular shift left by r_i)

For rounds 1, 2, 9, 16,

r_i is 1

For all other rounds,

r_i is 2

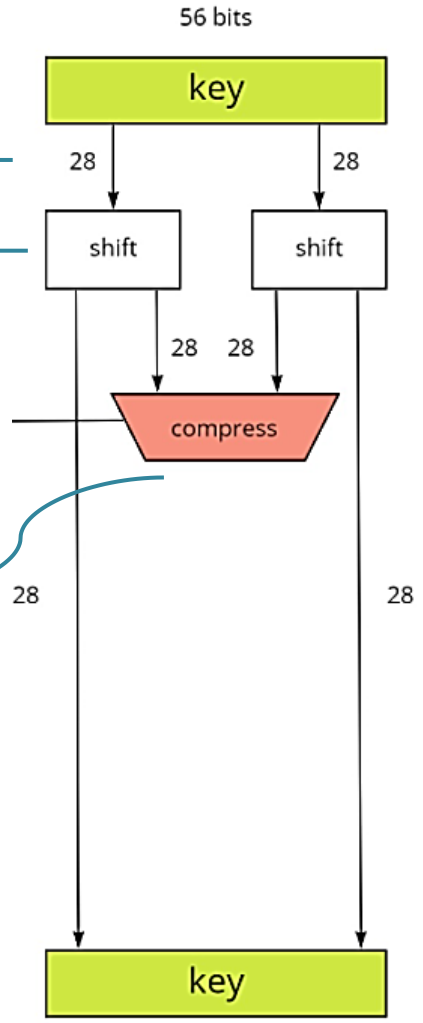
Left half of subkey K_i is of LK bits

13	16	10	23	0	4	2	27	14	5	20	9
22	18	11	3	25	7	15	6	26	19	12	1

Right half of subkey K_i is RK bits

12	23	2	8	18	26	1	11	22	16	4	19
15	20	10	27	5	24	17	13	21	7	0	3

(bits 8,17,21,24 of LK omitted each round
bits 6,9,14,25 of RK omitted each round)



DES 주요 특징

- DES의 구조 : Feistel cipher
  - ✓ 64 bit block length
  - ✓ 56 bit key length
  - ✓ 16 rounds
  - ✓ 48 bits of key used each round (subkey : 16개)
- 각 단계의 동작은 간단(for a block cipher)
- 암호 알고리즘의 안전성은 주로 “S-box”에 달렸음.

DES Security

- DES의 안전성은 S-box에 의해 결정
  - ✓ 다른 동작은 선형
- DES는 30년 동안 안전 했고 백도어는 없는 것으로 판명
- 어떠한 수학적 암호 해석 공격에도 안전
- 결론
  - ✓ DES is secure!
  - ✓ DES designers were ahead of the curve.
- 1998년 Crack이라는 컴퓨터($250,00 이하)로 56시간에 무너짐
- 결국 56 bit 키는 너무 짧아서 더 이상 암호에 사용할 수 없음
- 3DES, AES(Advanced Encryption Standard)으로 발전

2. 대칭 암호화 방식

2 트리플 DES 알고리즘

- DES의 복호화가 가능해짐에 따라 AES (Advanced Encryption Standard)가 나오기 전까지 **임시로 사용한 암호화 알고리즘**
- DES 알고리즘과 비슷하지만 암호화 및 복호화 과정에서 DES와 달리 2개의 암호화 키를 이용
- DES 알고리즘보다 암호화 강도가 2배 더 높아 오래 사용되지 못함

2. 대칭 암호화 방식

3 AES 알고리즘

- DES의 암호화 강도가 점점 약해지면서 새롭게 개발된 것
- 1997년에 NIST는 암호화 알고리즘을 다시 공모
- 향후 30년 정도 사용할 수 있는 보안성, **128비트 암호화 블록**, 다양한 키 길이를 갖출 것이라는 공모 조건
- 빈센트 레이먼, 요안 다에먼이 개발한 Rijndael 알고리즘이 2000년 10월 최종 AES 알고리즘으로 선정

4 SEED 알고리즘

- 전자상거래, 금융, 무선통신 등에서 전송되는 중요한 정보를 보호하기 위해 **한국인터넷진흥원과 국내 암호 전문가들이 순수 국내 기술로 개발한 128 비트 블록의 암호화 알고리즘**
- SEED 128은 1999년 9월 정보통신단체표준(TTA 표준)으로 제정되었고, 2005년에는 ISO/IEC와 IETF로부터 암호화 표준 알고리즘으로 인정
- 국내에서 개발된 많은 암호 프로그램과 보안 솔루션에서 사용

2. 대칭 암호화 방식

5 ARIA 알고리즘

- 전자정부 구현으로 다양한 환경에 적합한 암호화 알고리즘이 필요하여 국가보안기술연구소(NSRI) 주도로 개발
- 2004년 국가표준기본법에 의거하여 국가표준㉮으로 지정
- AES 알고리즘과 마찬가지로 128/192/256비트 암호화 키를 지원

6 양자 암호

- 1984년 찰스 베넷과 질 브라사르가 BB84라는 프로토콜을 통해 제안
- 광자 하나하나의 단위로 신호를 실어 나름, 편광이나 위상차를 이용하여 신호를 전송
- 양자역학의 복제 불가능성 원리와 측정 후 붕괴라는 특이한 현상을 이용함
- 단일 광자를 정확하게 측정할 수 있는 기회가 단 한 번으로 제한
- 차세대 암호화 기술로 많은 관심을 받고 있지만 광자의 특성상 가용 통신 거리가 최대 약 200km로 짧고 고가의 장비로 제한적인 범위에 사용

2. 대칭 암호화 방식

IDEA 알고리즘

- 128비트 키를 사용하여 64비트 평문을 8라운드를 거쳐 64비트 암호문으로 만드는 방식, 주로 키 교환에 쓰임

RC5 알고리즘

- 비교적 간단한 연산으로 빠른 암호화와 복호화 기능을 제공하며 **하드웨어에 적합함, DES의 약 10배**

Skipjack 알고리즘

- 미국 국가안보국(NSA)에서 개발한 클리퍼 칩에 내장된 블록 알고리즘, 공개, 64비트 입출력, 80비트 키를 이용하여 총 **32라운드의 암호화 과정을 수행**

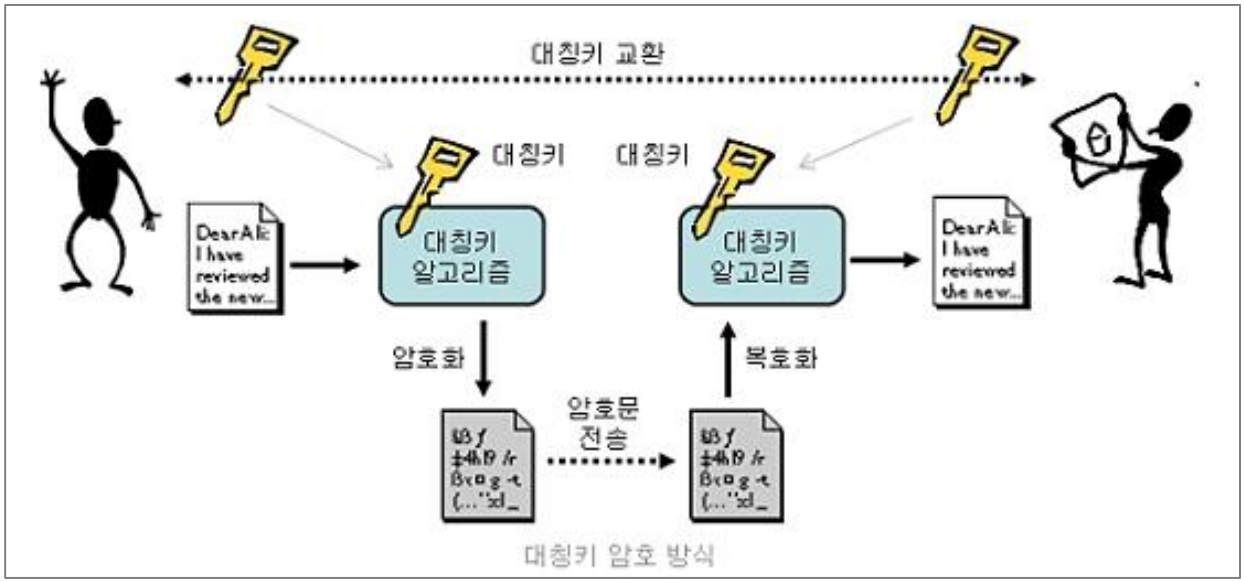
LEA 알고리즘

- 고속 환경 및 **모바일 기기 등의 경량 환경에서 기밀성을 제공**, 국내에서 개발된 암호화 알고리즘,128비트 데이터 블록으로 128, 192, 256비트 비밀 키를 사용 가능

3. 비대칭 암호화 방식

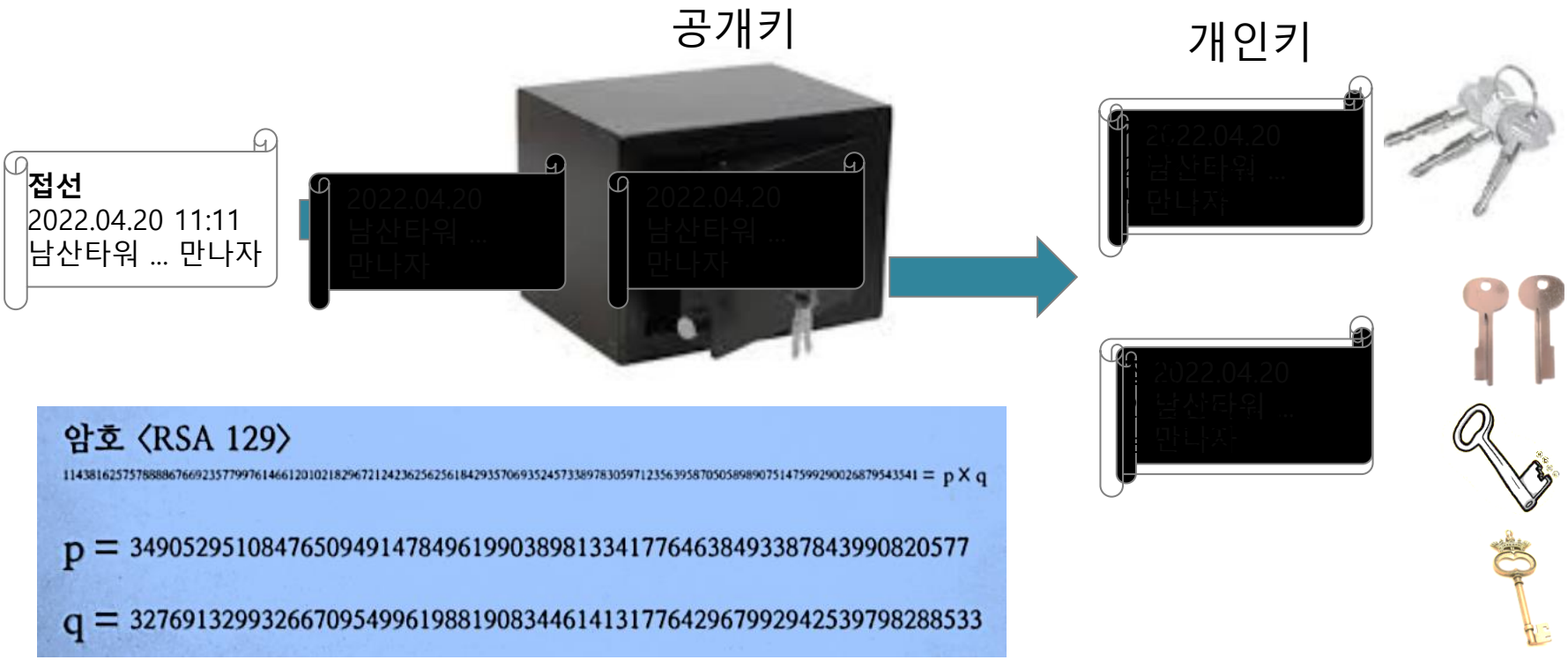
대칭키 암호화 방식의 약점

- AES 알고리즘이 개발되면서 암호 해독이 거이 불가능한 알고리즘으로 인정
- 대칭 암호화 방식으로 암호화하면 복호화하는 사람도 암호화 Key 필요
- 따라서 암호화 Key를 전달해야 함
- 암호문과 Key를 함께 보내면 암호화한 의미 없음
- 암호문과 Key를 따로 전달해야 함
- 인터넷이 활성화된 현대에는 암호문을 직접 전달할 방법이 없음.



기본 원리(개념)

91을 소인수분해하면 7과 13이 된다. 7과 13을 주면 곱해서 91을 금세 만들 수 있지만, 91을 주고 소인수분해해 7과 13을 찾아내려면 시간이 걸린다. 이렇게 **한 쪽 방향의 계산은 쉬운데 역방향 계산은 어려운 원리를 이용해 암호를 만든다.”**



암호 〈RSA 129〉

1143816257578888676692357799761466120102182967212423625625618429357069352457338978305971235639587050589890751475992900268795435 41 = p X q

p = 3490529510847650949147849619903898133417764638493387843990820577

q = 32769132993266709549961988190834461413177642967992942539798288533

3. 비대칭 암호화 방식

1 비대칭 암호화 방식의 발견

휫필드 디피 - 해시넷

마틴 헬만 - 해시넷

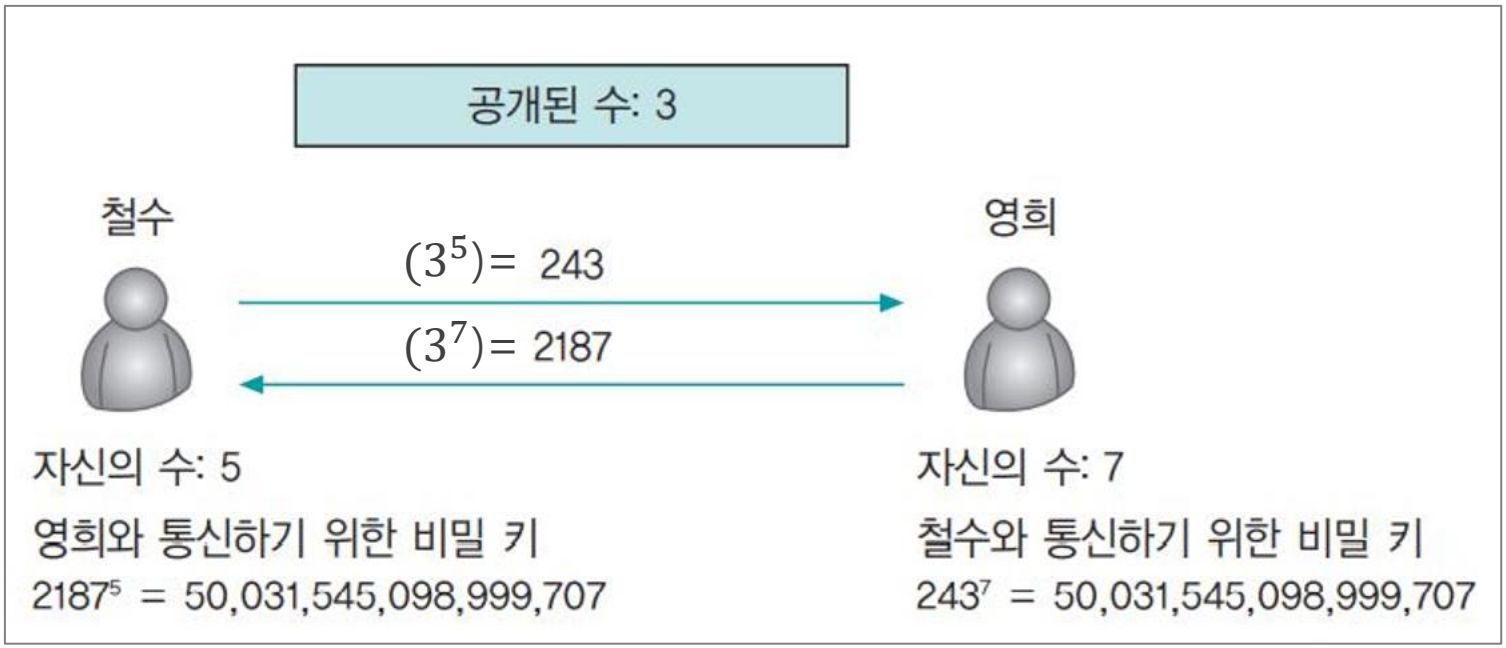
암호 〈RSA 129〉

1143816257578888676692357799761466120102182967212423625625618429357069352457338978305971235639587050589890751475992900268795435 41 = p X q

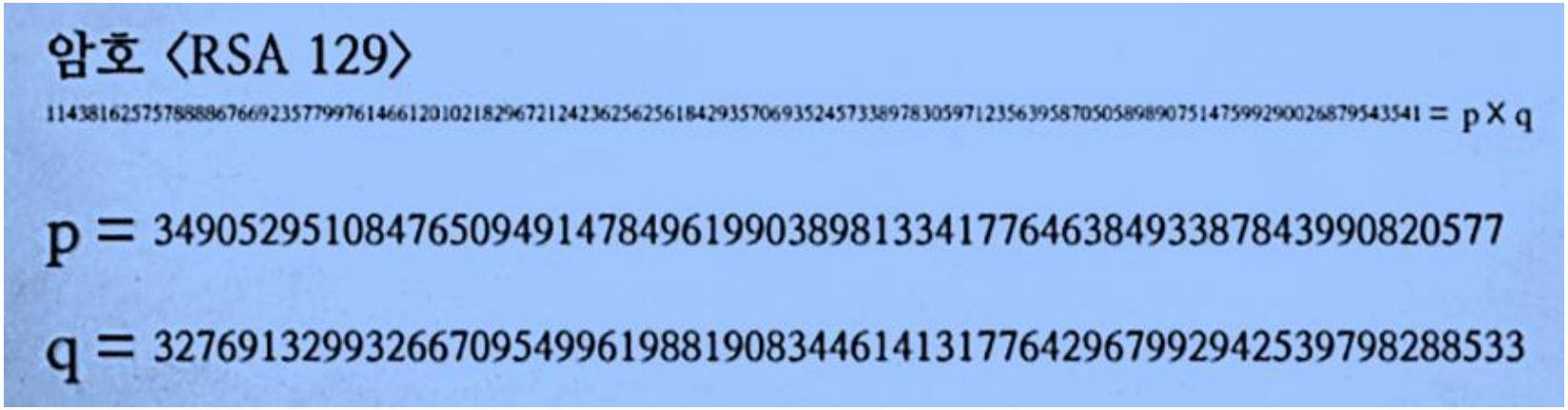
p = 3490529510847650949147849619903898133417764638493387843990820577

q = 32769132993266709549961988190834461413177642967992942539798288533

RSA-2048



3. 비대칭 암호화 방식



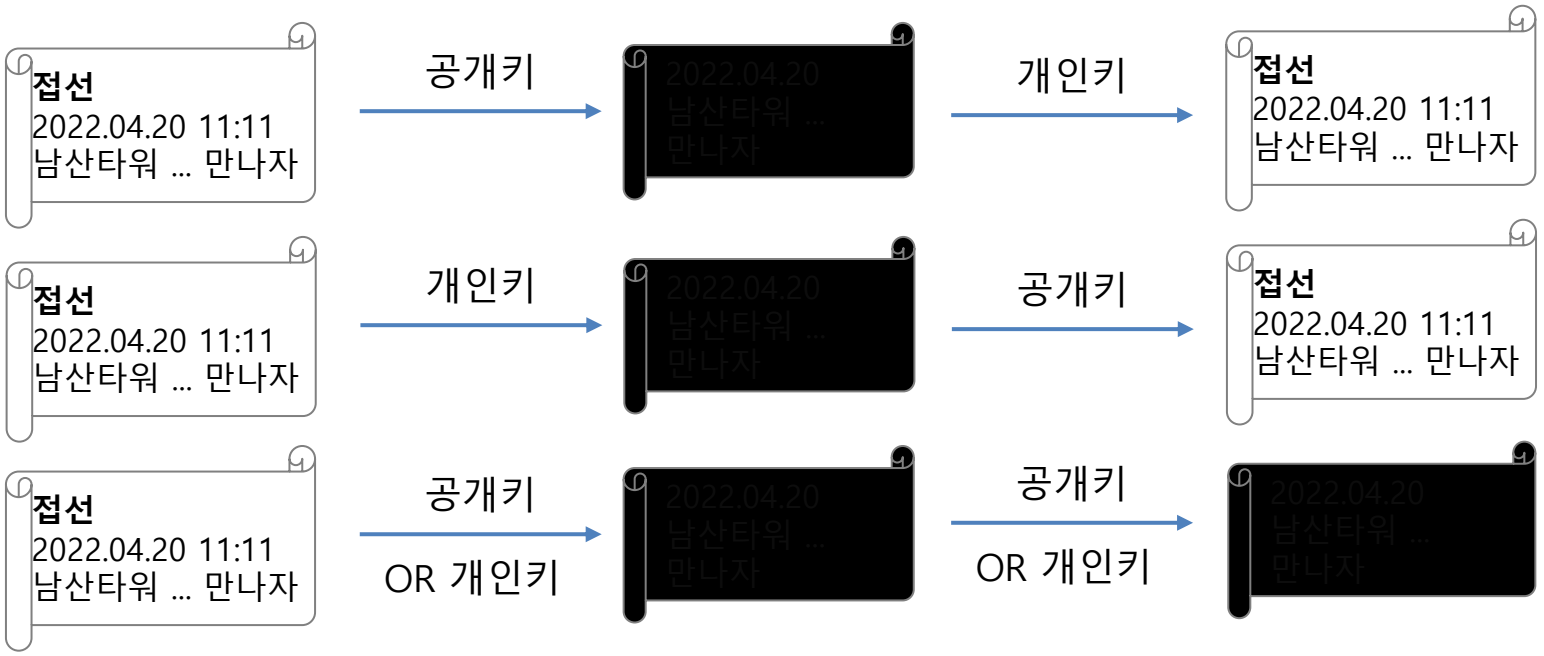
- 1977년에 미국 과학 전문지인 사이언티픽 아메리칸(Scientific American)에 'RSA129' 숫자를 공개하고 현상금을 걸었다.
- 64자리의 소수 2개를 곱해서 129자리 숫자를 만들었는데, 17년 뒤인 1994년에 25개국 600명이 1600대의 컴퓨터를 동원하여 8개월 만에 풀어냈다.
- 만약 **2000자리 숫자를 소인수 분해하려면** 우주에 있는 양성자와 중성자 등 모든 핵입자의 수인 10의 80제곱만큼의 수퍼 컴퓨터를 동원해서, 우주의 나이인 138억년, 즉 10의 18제곱초 동안 계산하더라도 불가능하다고 이야기하는 사람도 있다."

3. 비대칭 암호화 방식

- 비대칭 암호화 알고리즘 중에서 가장 많은 지지를 받으며 오늘날 산업 표준
- 알고리즘은 MIT의 로널드 리베스트, 아디 샤미르, 레너드 애들먼이 고안
- RSA 암호는 기본적인 정수론, 즉 소수(素數)를 이용
- 중요 정보를 소수 2개로 표현한 후 두 소수의 곱을 힌트와 함께 전송하여 암호로 사용하는 것
- **RSA 알고리즘에서는 모든 사람이 고유한 N 값(두 소수 의 곱)을 가짐**
  - ✓ 영희가 p=17,159와 q=10,247의 곱을 자신의 N 값(17,159×10,247= **175,828,273**)으로 정함
  - ✓ 영희의 공개 키인 N 값이 모든 사람에게 공개
  - ✓ 영희에게 메시지를 보내고 싶은 사람은 N 값을 이용하여 보내는 메시지를 어떤 알고리즘으로 암호화한 후 영희에게 전송
  - ✓ 여기서 영희의 **개인 키는 p와 q**

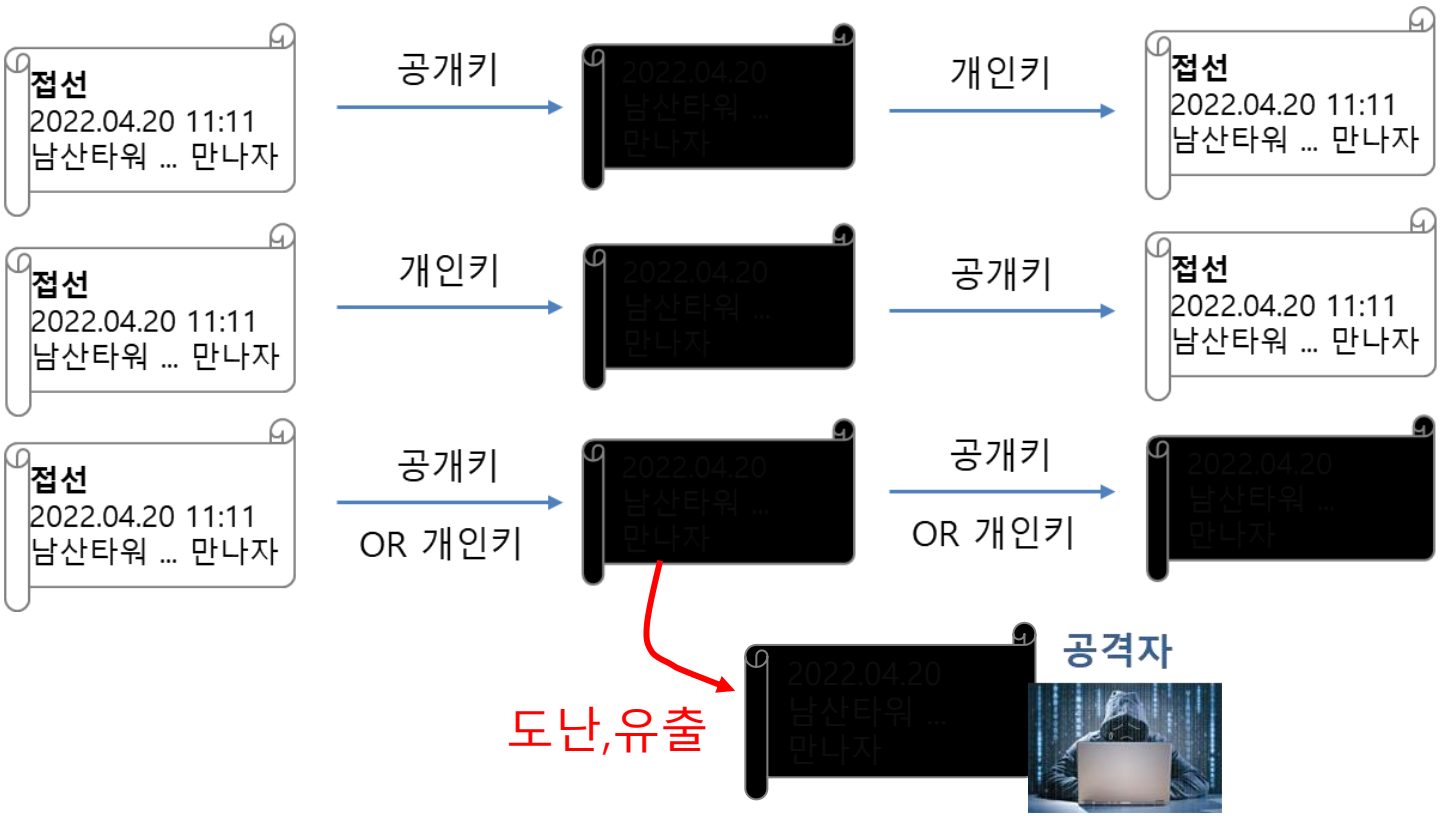
3. 비대칭 암호화 방식

- 비대칭 암호화 알고리즘은 RSA 알고리즘이 나오면서 정립
- 각 개인이 공개 키와 개인 키를 소유하는 구조지만 서로의 개인 키는 얻을 수 없음
- 대칭 암호화 알고리즘과 달리 메시지의 암호화와 복호화가 같은 키로 이루어지지 않음
- 언제나 한 쌍의 개인 키와 공개 키로 암호화와 복호화가 이루어짐



3. 비대칭 암호화 방식

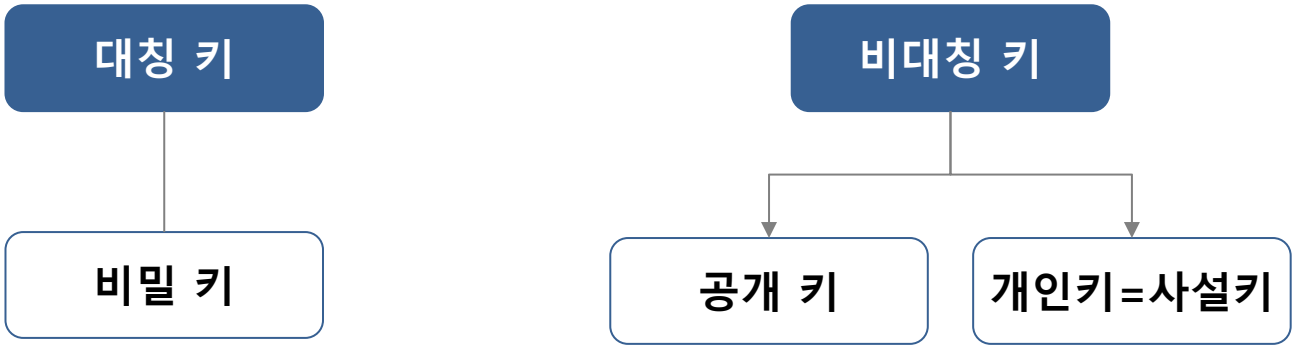
- 기밀성
- 부인방지 기능



3. 비대칭 암호화 방식

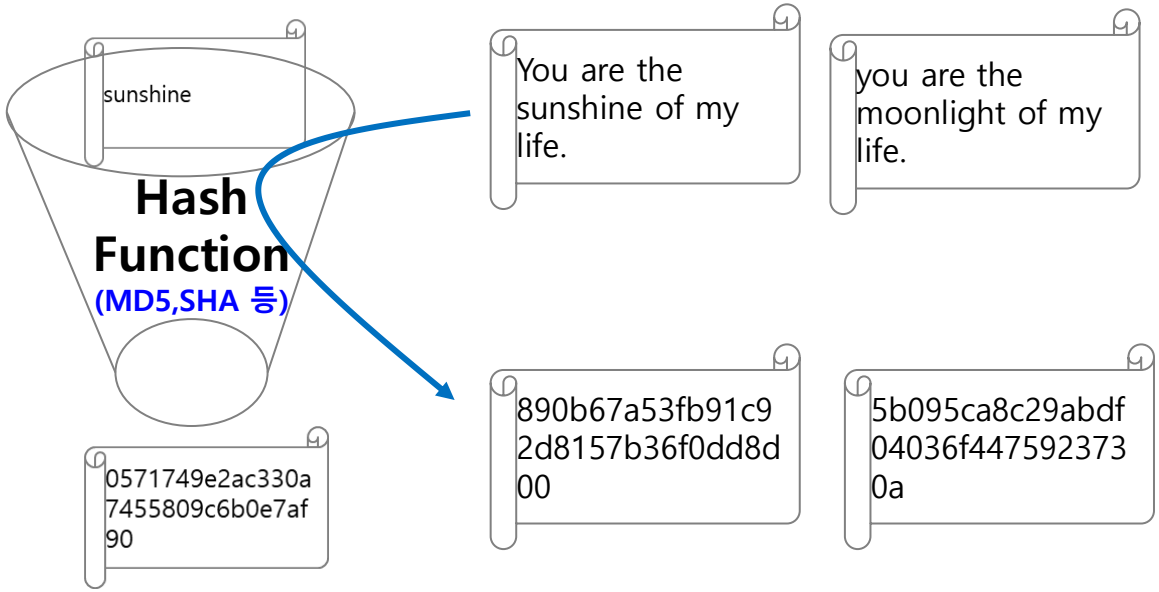
비대칭 암호화의 기능

- 대칭 키: 암호화할 때 쓰는 키와 복호화할 때 쓰는 키가 같은 것
- 비밀 키: 암호화할 때와 복호화할 때 사용되는 키가 같으므로 암호문이 효력을 발휘하려면 발신자와 수신자 사이의 키에 대한 정보가 비밀로 유지
- 비대칭 키: 암호화할 때 쓰는 키와 복호화할 때 쓰는 키가 다른 것 (공개 키와 개인 키를 묶어 비대칭 키)
- 공개 키와 개인 키: 발신자와 수신자가 각각 한 쌍을 소유



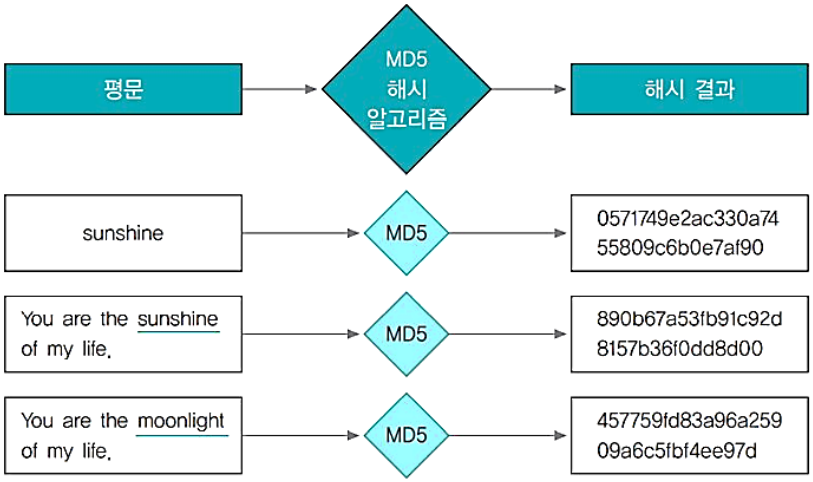
4. 해시(Hash)

- **하나의 문자열을 이를 뜻하는 더 짧은 길이의 값이나 키로 변환하는 것**.
- 암호화는 정보를 숨기기 위한 장치
- **해시는 정보의 위조·변조를 확인하기 위한 것**, 즉 정보의 무결성을 확인하기 위한 것
- 대칭 암호화, 비대칭 암호화, 해시 기법을 사용하여 전자서명, 전자봉투, 전자화폐 등 다양한 전자상거래 기능 구현 가능



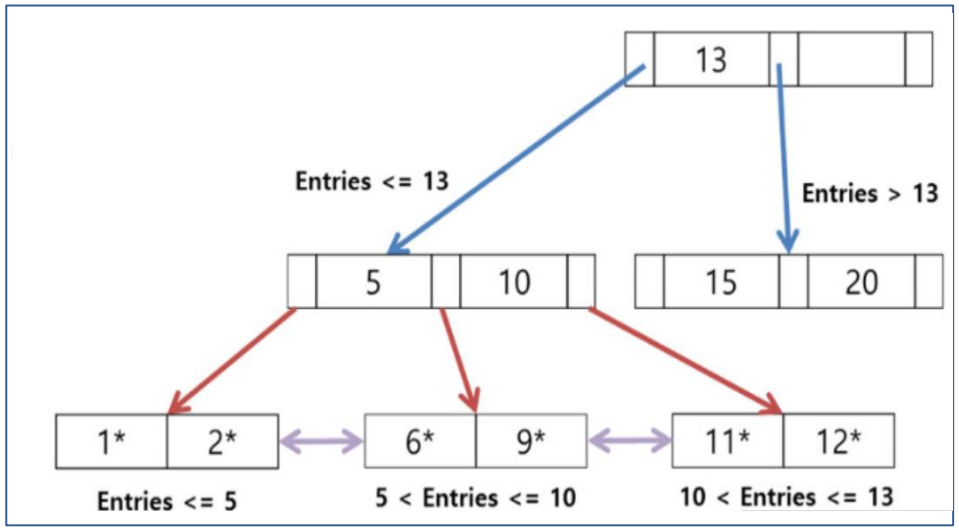
4. 해시(Hash)

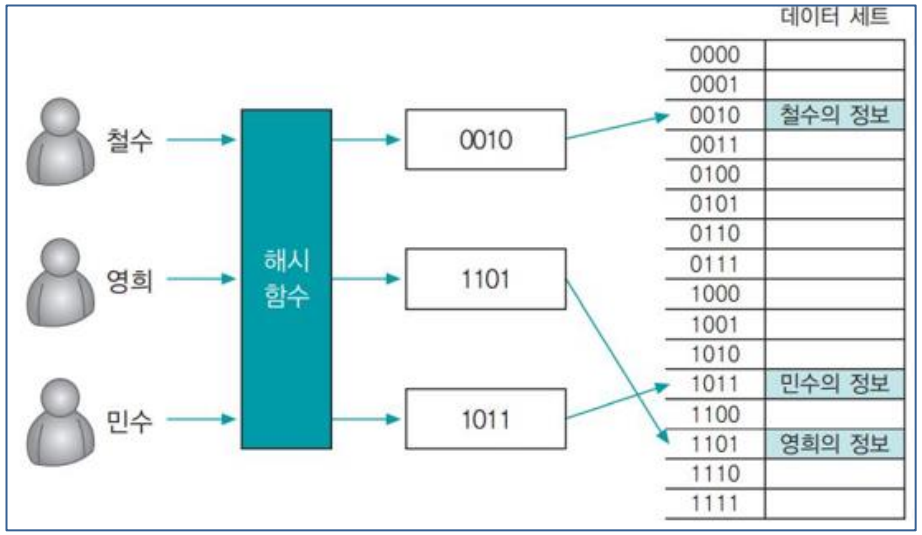
- 해시는 입력되는 세개의 평문은 길이가 각각 다르지만 해시 결과는 32개 문자로 길이가 모두 같음
- 두 번째와 세 번째 평문은 단어 하나만 다를 뿐인데 해시 결과는 완전히 다름
- 해시되기 전의 값을 해시 값으로 추측하기가 불가능하다는 특징 때문에 일어난 결과
- **충돌**: 다른 값의 데이터를 입력하더라도 해시 결과 값이 같을 수 있음.
- MD5는 32개의 16진수로 이루어 졌으므로 16^{32}개 만들 수 있음
- 340,282,366,9,920,938,463,463,374,607,431,768,211,456개 만들 수 있음.
- 거의 무한대



4. 해시(Hash)

- 원래 해시는 데이터베이스의 탐색을 효과적으로 구현하기 위해 만들어진 것
- 보안에서는 해시가 완전히 똑같은 데이터만 해시 값이 같고 조금만 달라도 해시 값이 전혀 다르다는 점을 이용하여 데이터가 임의로 변경되지 않았다는 데이터 무결성을 확인하기 위한 도구로 사용.





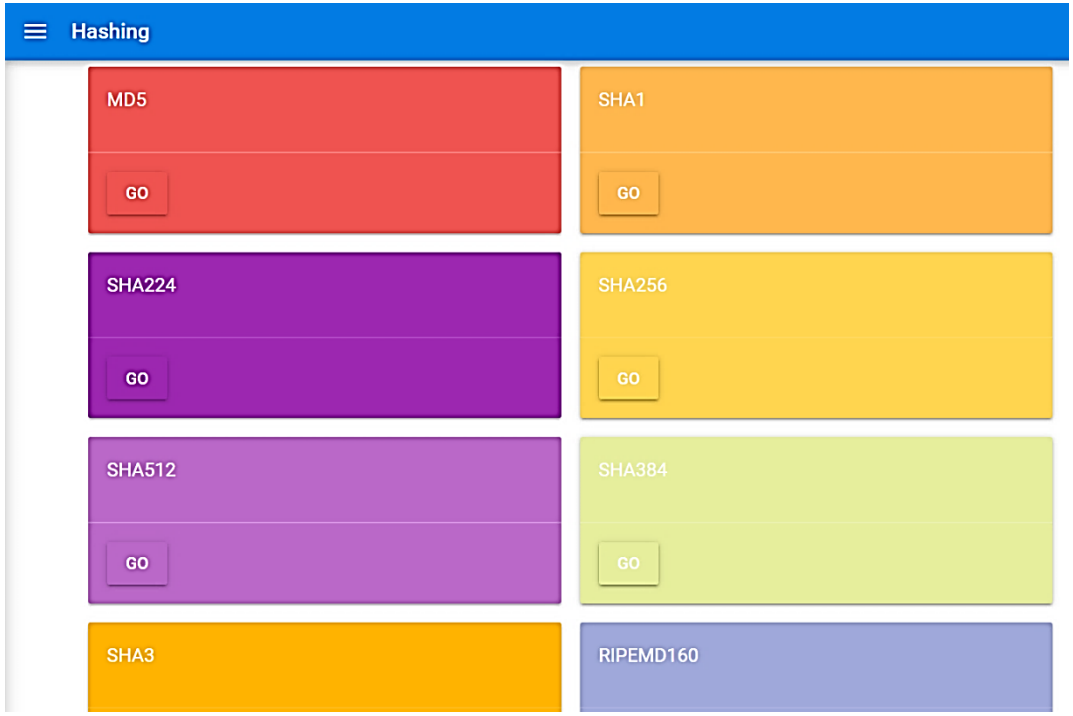
DB에서 해시 값을 통한 값의 참조

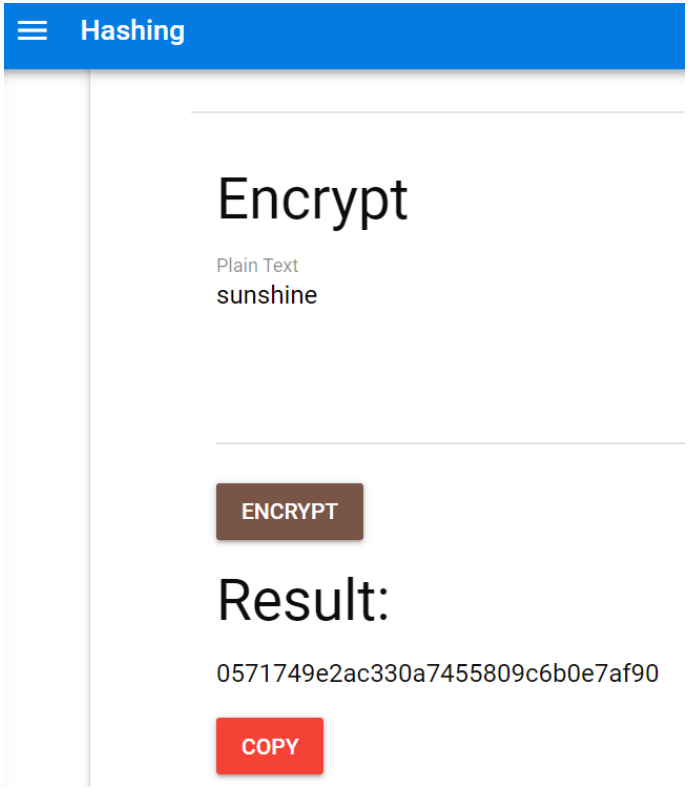
4. 해시(Hash)

해시 값 구하기

https://aes.kr/#/

https://www.convertstring.com/ko/Hash/MD5





정보보안개론
c202f1cb53176cdf3b91e9c515385e9e
C202F1CB53176CDF3B91E9C515385E9E
정보보안개론.
2d299fbe1f640f201aa6f1872d657958

4. 해시(Hash)

대표적인 해시 알고리즘은 MD와 SHA가 있음

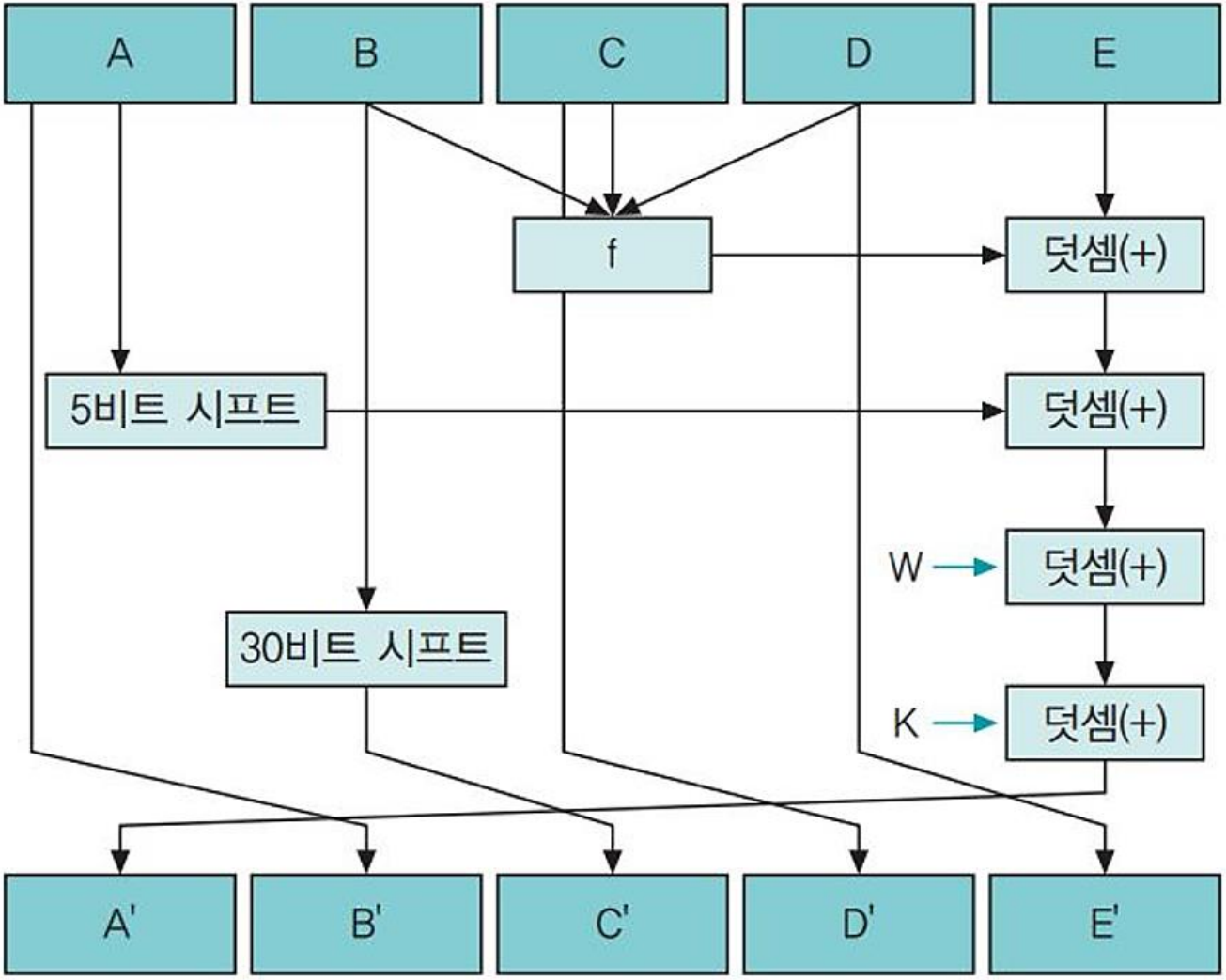
MD 알고리즘(Message Digest Function 95)

- 로널드 리베스트가 **공개 키 기반 구조**를 만들기 위해 RSA와 함께 개발한 것으로 MD2, MD4, MD5가 있음
- 1989년에 개발된 MD2는 8비트 컴퓨터에 최적화
- 1990년에 개발된 MD4와 1991년에 개발된 MD5는 32비트 컴퓨터에 최적화
- MD5 알고리즘은 MD4의 확장판으로 MD4보다 보안성이 더 뛰어남

SHA

- 미국 국가보안국이 만든 SHA(Secure Hash Algorithm)는 160비트 값 생성.
- 160비트의 값을 생성하는 해시 함수로, MD4가 발전한 형태
- MD5보다 조금 느리지만 좀 더 안전하다고 알려져 있으며 SHA에 입력하는 데이터는 512비트 크기의 블록임
- SHA 알고리즘은 크게 SHA-1과 SHA-2로 나눌 수 있음.

SHA의 동작 원리



SHA의 종류와 특징

표 7-1 SHA 종류와 특징

알고리즘		해시 값 크기	내부 상태 크기	블록 크기	길이 한계	워드 크기	과정 수	사용되는 연산	충돌
SHA-0		160	160	512	64	32	80	+, and, or, xor, rotl	발견
SHA-1		160	160	512	64	32	80	+, and, or, xor, rotl	발견
SHA-2	SHA-224	224	256	512	64	32	64	+, and, or, xor, shr, rotr	-
	SHA-256	256	256	512	64	32	64	+, and, or, xor, shr, rotr	-
	SHA-384	384	512	1024	128	64	80	+, and, or, xor, shr, rotr	-
	SHA-512	512	512	1024	128	64	80	+, and, or, xor, shr, rotr	-

SHA와 MD5의 활용
TLS, SSL, PGP, SSH, S/MIME, IPsec 등에서 활용

Thank you