DES

김현지

https://youtu.be/7W64z7bG0BQ

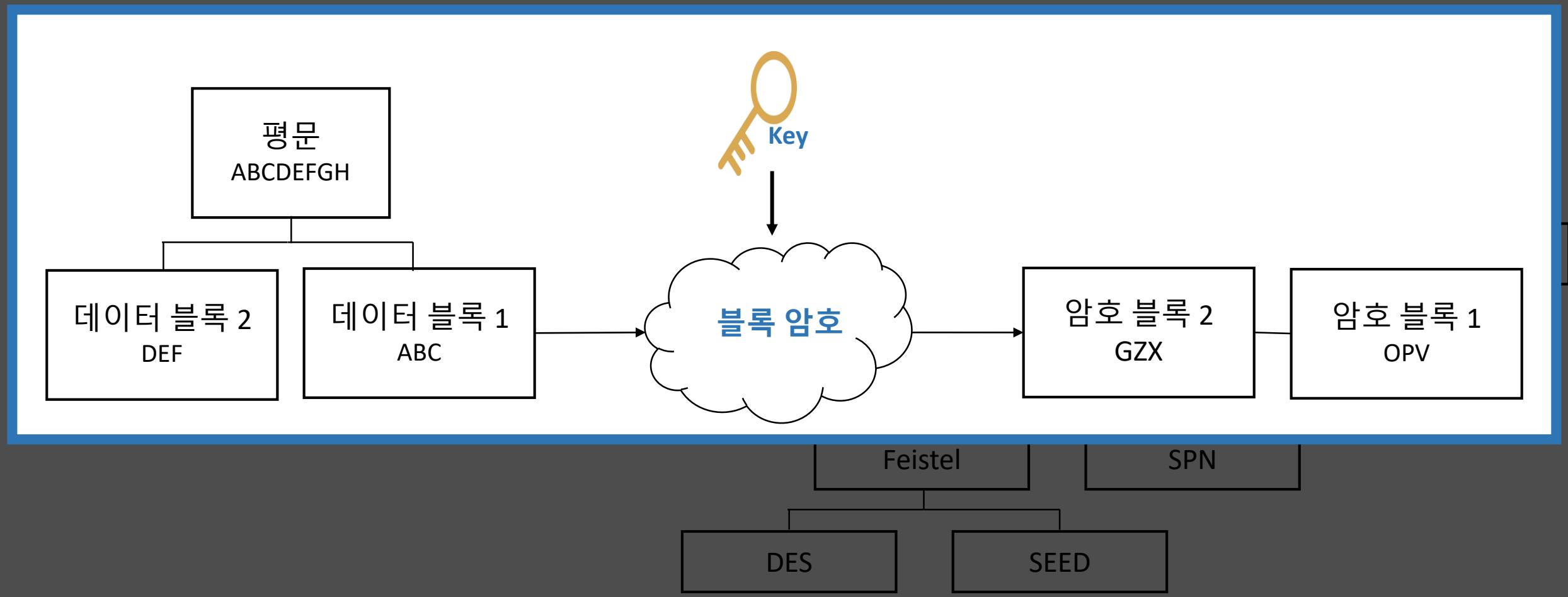
Contents

CryptoCraft LAB

01. Block cipher

01. Block cipher

❖ 특정 길이의 n-bit 블록을 n –bit 블록으로 변환



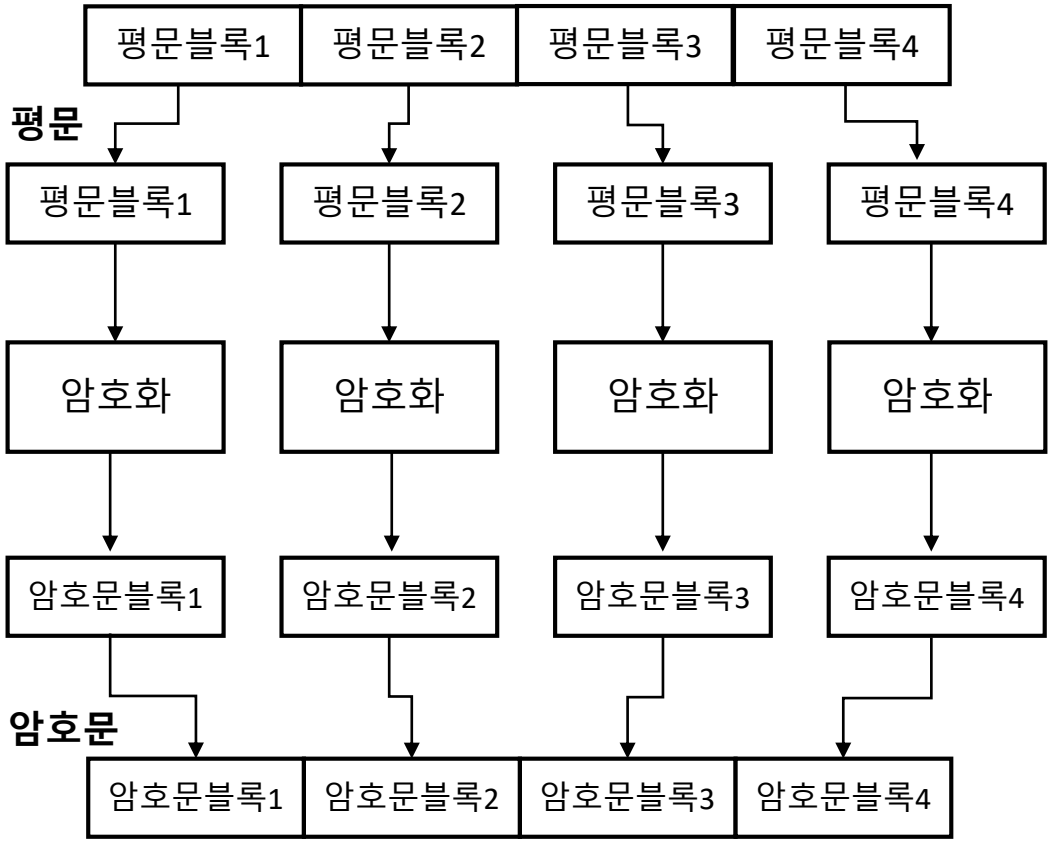
01. Block cipher

❖ Feistel vs SPN

	Feistel		SPN
Encryption & Decryption	같음		다름
반복 연산 효과	2-round	=	1-round
1-round 당 암호화 bit 수	32 bit 씩		128 bit 씩
병렬성	낮음		높음
대표 알고리즘	DES		AES

01. Block cipher

❖ ECB

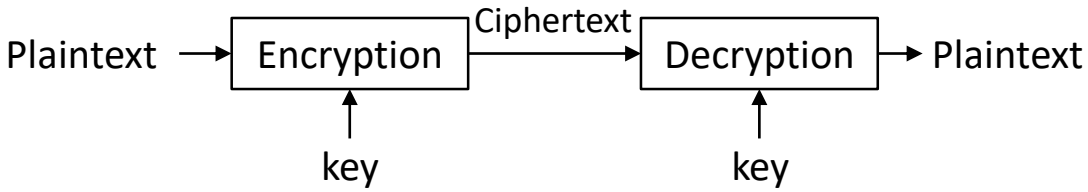


❖ 장점

- 블록동기화 필요 x
- 비트에러 발생 시에도 해당 블록에만 영향
- 병렬화 x : 고속구현에 적절

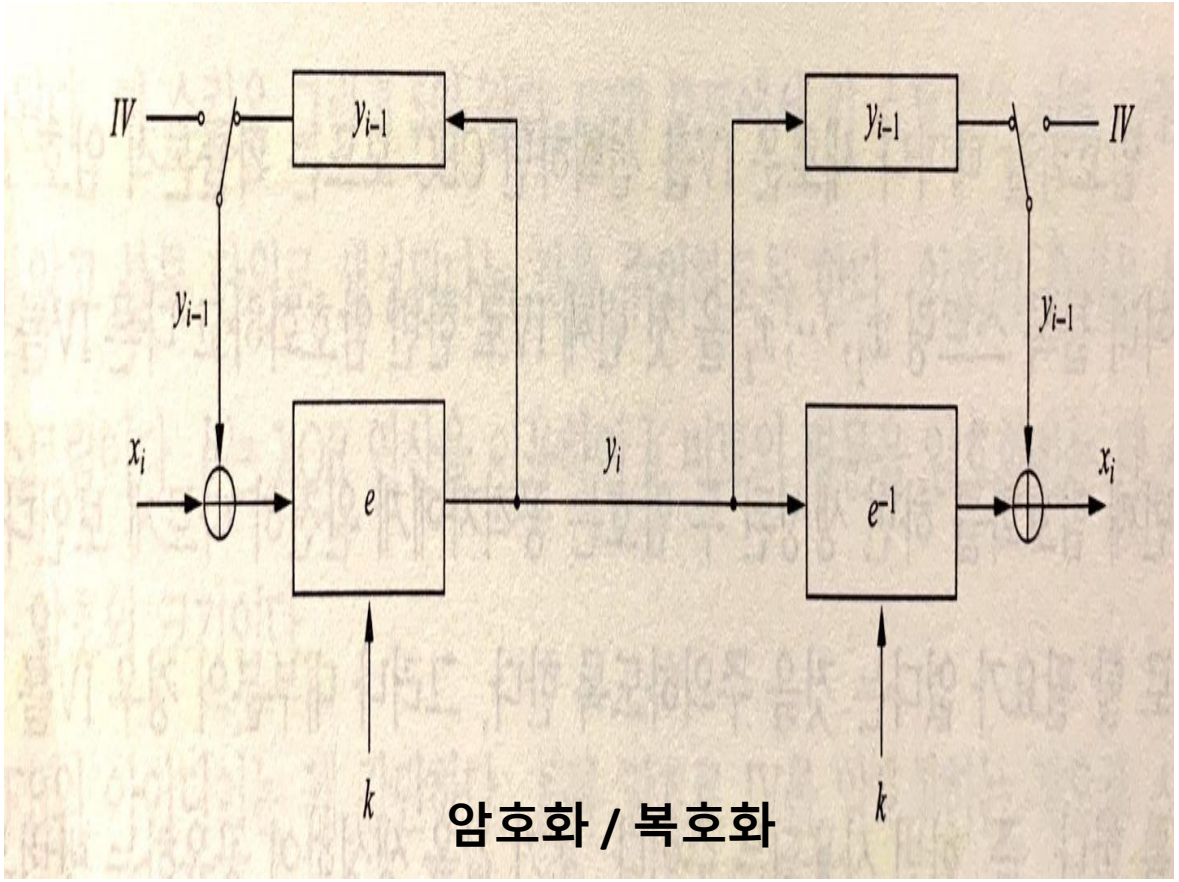
❖ 단점

- 결정론적 암호화
- 트래픽분석 가능
- 이전 블록과 독립적으로 암호화
- 대체공격에 취약



01. Block cipher

❖ CBC



암호화 / 복호화

❖ Idea

- 모든 블록의 암호가 연결 : 암호문은 이전 모든 평문 블록따라 다름

  → 비결정론적

- Initial vector 이용 → random

➢ 블록 내 평문 비트의 암호는 동일한 블록 내 다른 평문에 의존

❖ 암호화 / 복호화

- 첫 블록 (i=1)

  → $Y_1 = e_K(X_1 \oplus IV)$ / $X_1 = e_K^{-1}(Y_1) \oplus IV$

- 그 외 (i>=2)

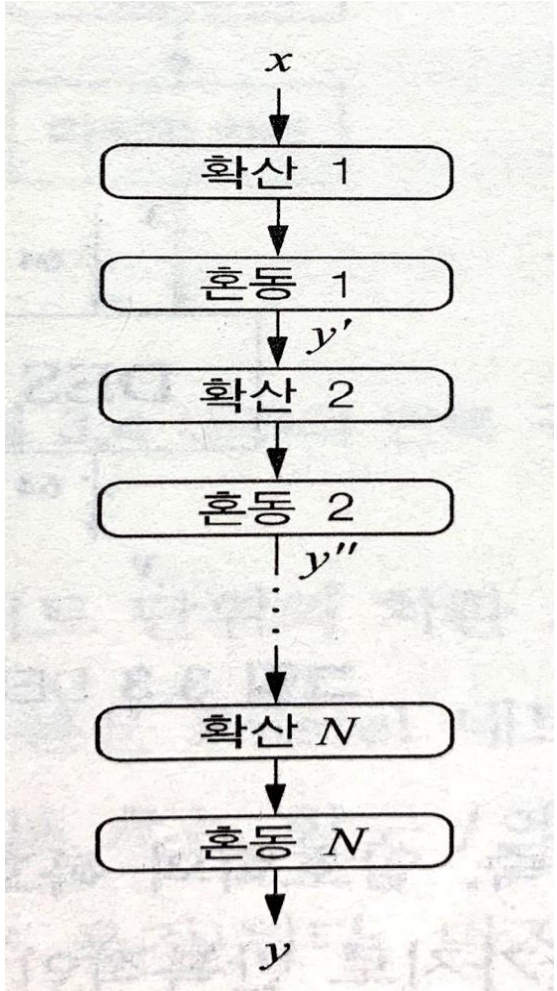
  → $Y_i = e_K(X_i \oplus Y_{i-1})$ / $X_i = e_K^{-1}(Y_i) \oplus Y_{i-1}$

02. Product cipher

02. Product cipher

❖ Product cipher (곱암호)

- 혼동과 확산을 함께 사용 시, 안전한 암호의 설계가 가능하다고 제안
- 각 round는 혼동과 확산 연산 수행 : S-box & P-box
- Round가 반복되는 구조
- DES도 곱암호 구조



N-round product cipher

02. Product cipher : Confusion

❖ Confusion(혼동)

- 암호문 – 키 사이의 관계를 숨김
- 키의 단일 비트가 변하면 암호문의 거의 모든 비트가 변함

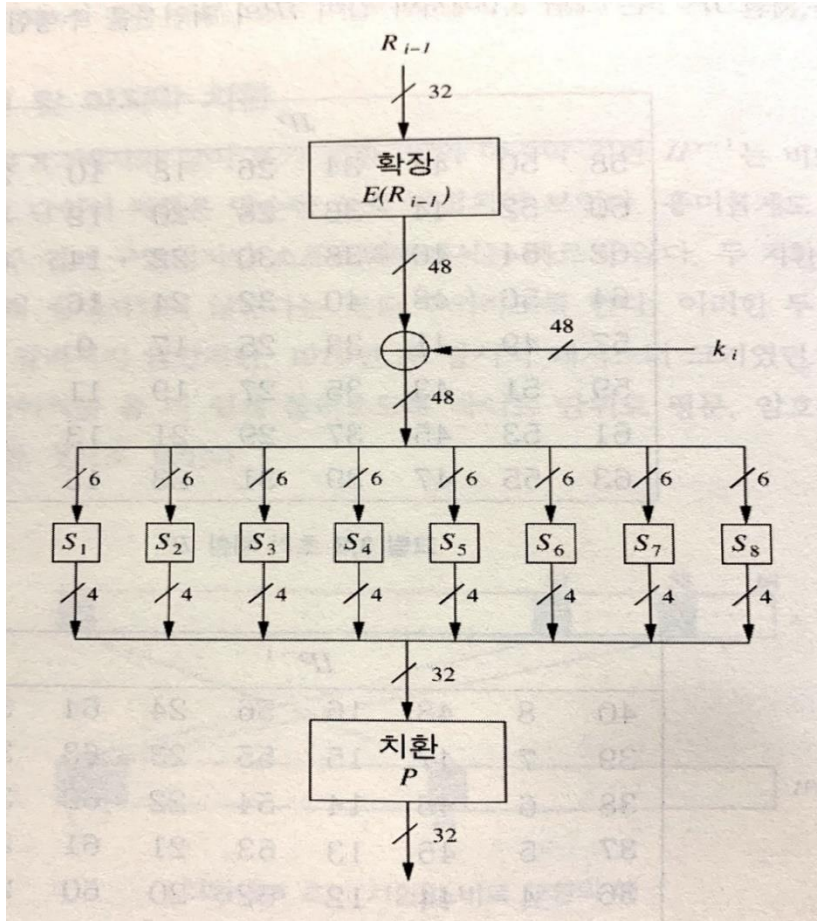
02. Product cipher : Diffusion

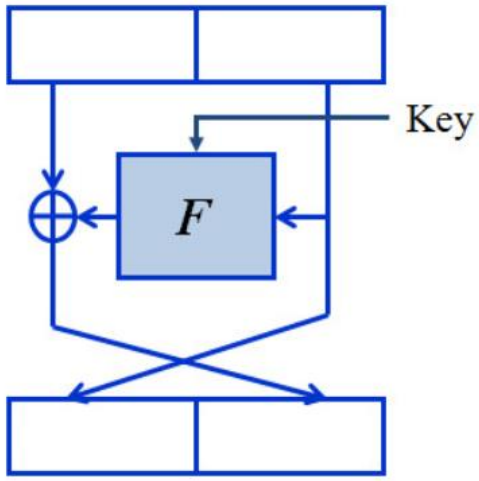
❖ Diffusion (확산)

- 암호문 - 평문 사이의 관계를 숨김
- 하나의 평문 기호가 많은 암호문 기호에 영향을 미치도록 하는 암호 연산
- 평문의 통계적 특성을 숨기는 것이 목적
- DES (비트 치환), AES (Mixcolumn 연산)

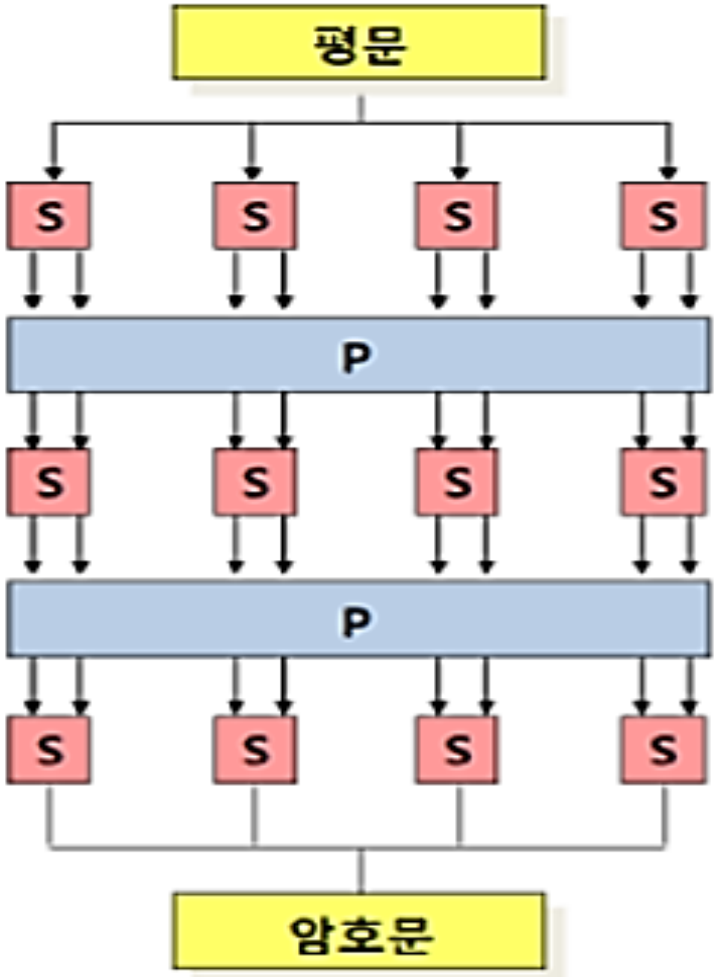
02. Product cipher

❖ Feistel





❖SPN



02. Product cipher

❖ Feistel 네트워크 구조

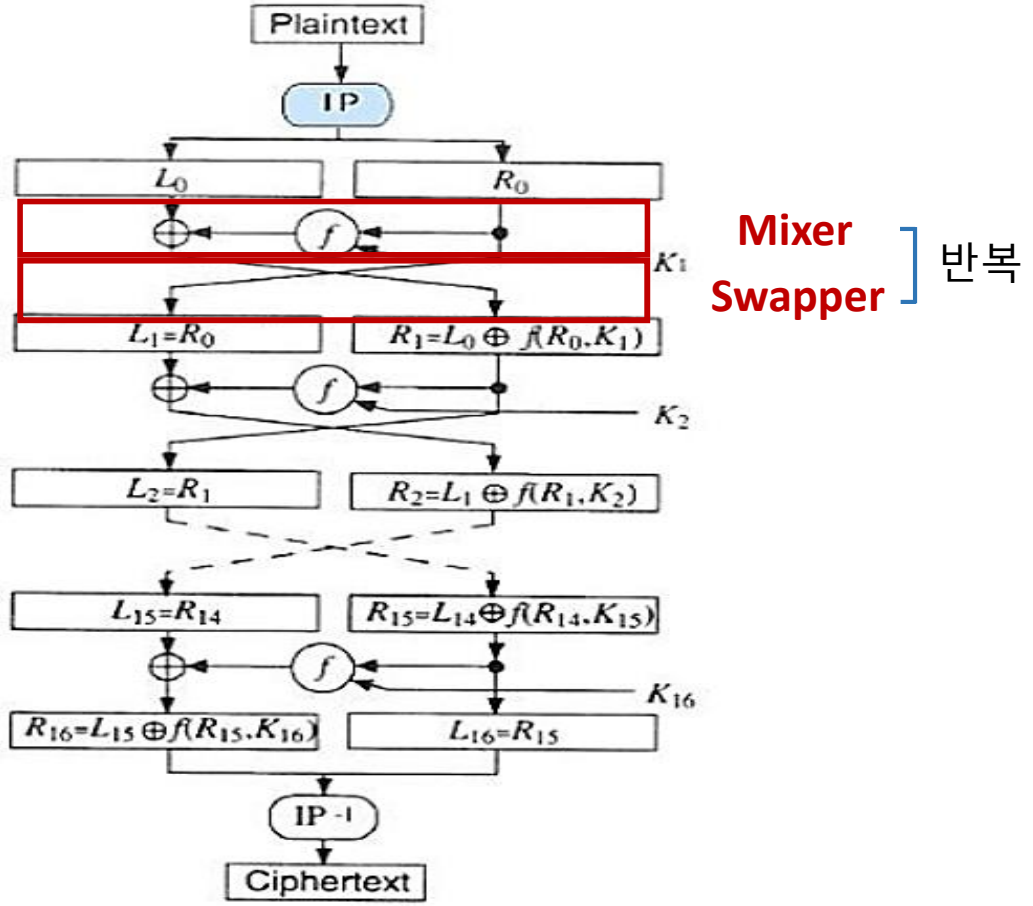
- Mixer

  $L_{i-1} \oplus f(R_{i-1}, K_i)$

- Swapper

  $L_{i-1} \oplus f(R_{i-1}, K_i)$ → R_i으로

  R_{i-1} → L_1으로

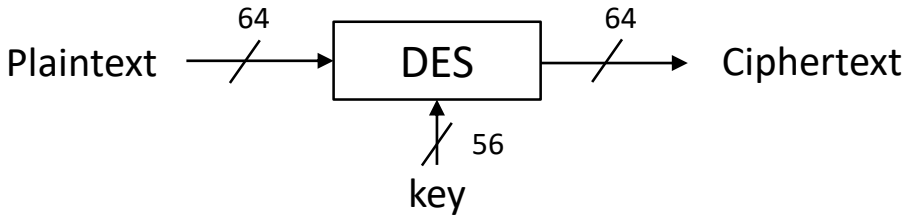


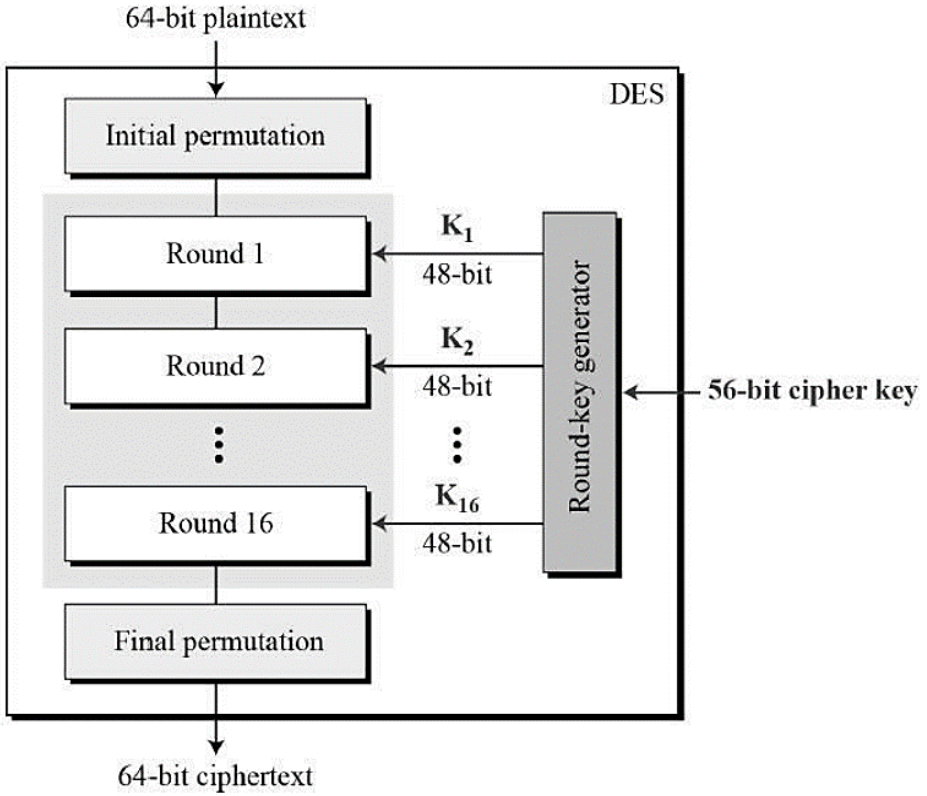
03. DES

03. DES (Data Encryption Standard)

❖ 구조

	DES
Block Size	64 bits
Key Length	56 bits
Round	16 rounds
Structure	Feistel



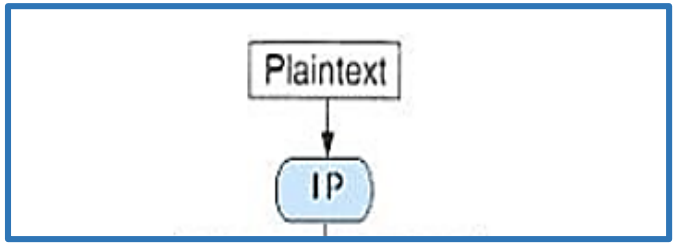


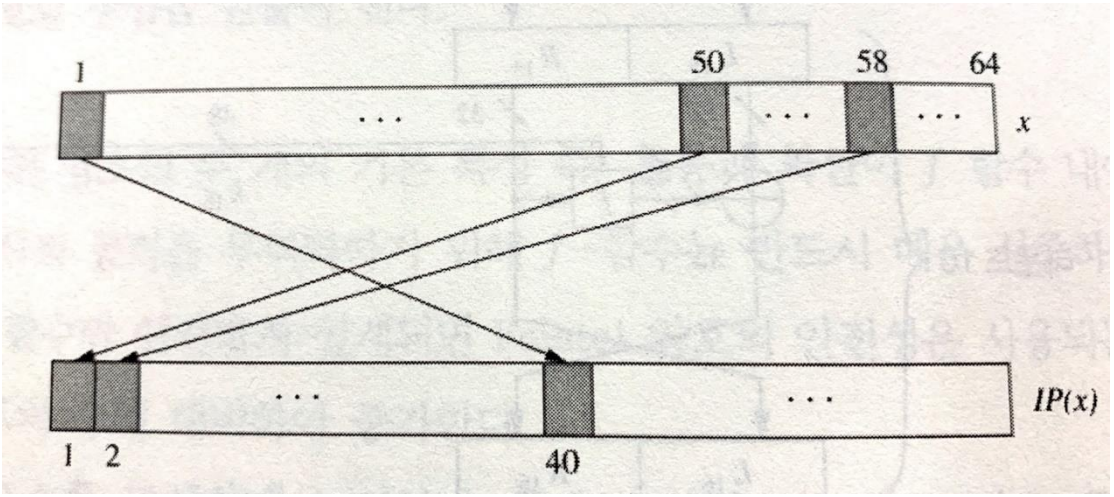
P-box 2개 : IP, IP^{-1}
feistel round function 16개
Round-key generator

03. DES (Data Encryption Standard)

① 64 비트의 평문 X는 Initial Permutation 후, 반으로 나뉨 → L_0,R_0

→ DES의 안전성 증대에는 영향 X





Initial permutation 비트 교환 예시

IP							
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	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Initial permutation table

✓ f – 함수

❖ S-box

① DES에서 유일한 비선형 함수

② 입력값 1 비트 상이 → 출력값은 2 비트 이상 상이

③ 암호학적 강도 측면에서 DES의 핵심

→ 혼동(Confusion) 에 관여

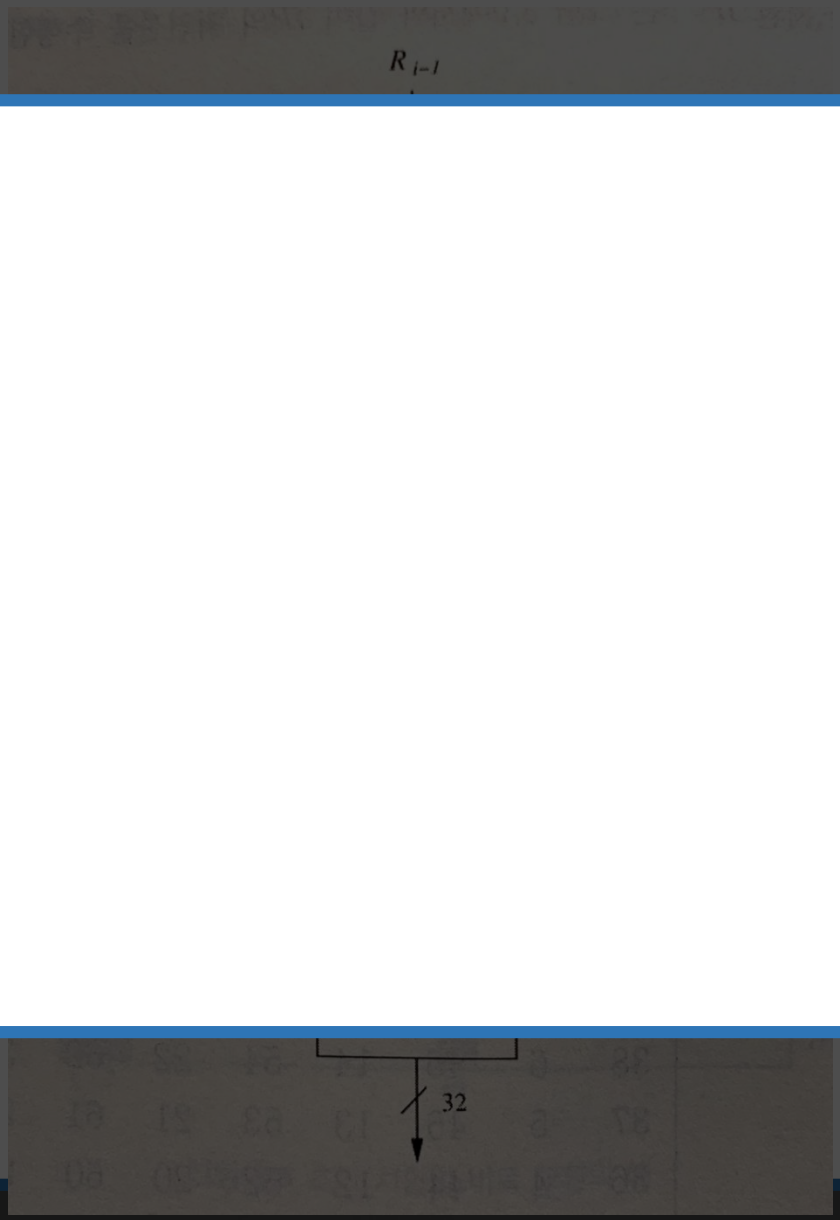
❖ P-box

① 확장 P-box 1개 : 32비트 → 48비트

② 단순 P-box 1개 : 32비트 → 32비트

→ 확산(Diffusion) 에 관여

➢ F-함수 결과 : L_{i-1} 의 암호화 위해 XOR 마스크로 사용



03. DES (Data Encryption Standard)

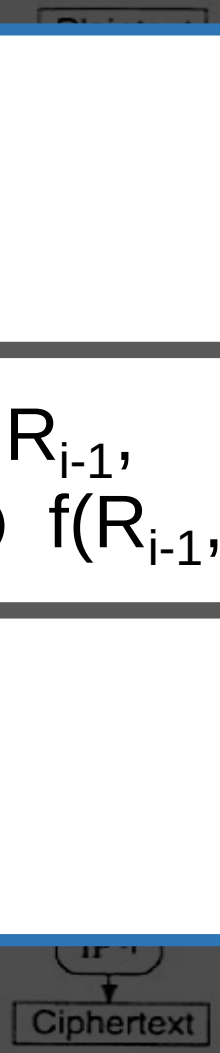
1 round라고 할 때, (i=1)

F-함수의 결과 값 $\oplus$ L_0 → R_1 으로 감

R_0 → L_1로 감

각 round마다 입력 비트의 왼쪽 L_{i-1}만 암호화

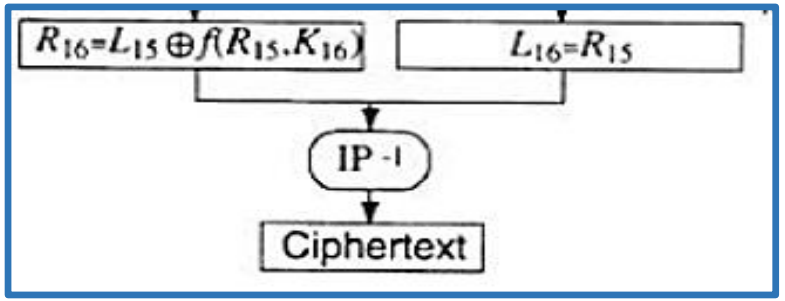
$$L_i = R_{i-1},$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

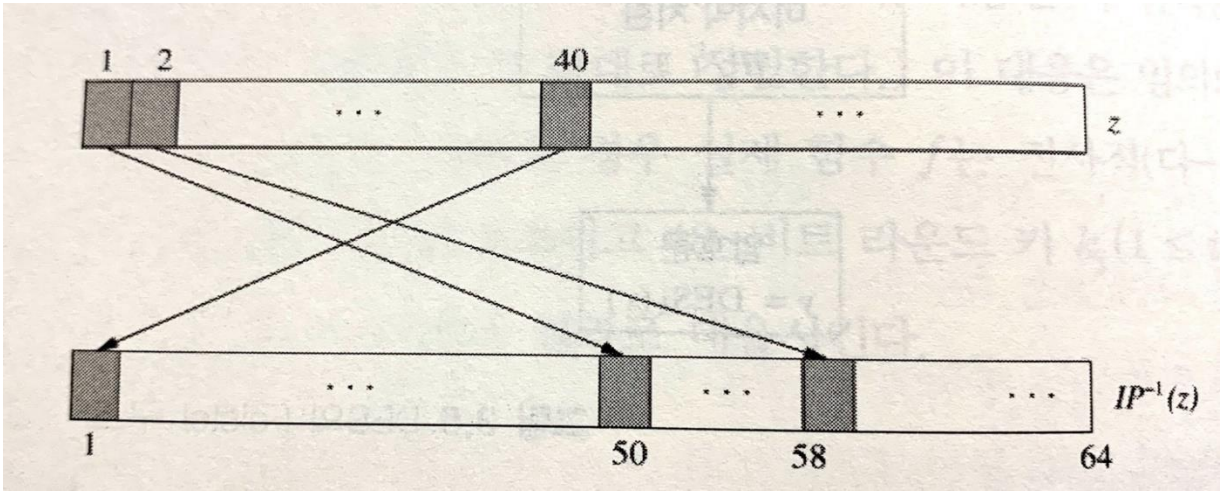


03. DES (Data Encryption Standard)

⑦ 16 round까지 진행 후, L_{16} , R_{16} 을 교환 후 Final Permutation

→ Initial Permutation의 역연산





Final permutation 비트 교환 예시

IP^{-1}

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

Final permutation : IP^{-1}

03. DES (Data Encryption Standard)

❖ KEY

- 56비트의 비밀키
- Round-key generator 에 의해 56비트짜리 주 키(K)에서 16개의 서로 다른 서브키(K_i) 생성

  → 각 round에서는 48비트의 서로 다른 서브키 사용
- 각 56 bit key가 서로 다른 round-key 에서 사용되도록 설계

  → 각 bit는 거의 16개중 14개 round-key 에서 사용
- 복호화 = 암호화 (feistel 구조)

  → 반대의 키스케줄링만 필요 → HW 구현에서의 장점

03. DES (Data Encryption Standard)

❖ 키 스케줄

- 원래의 56 비트 키로부터 48 비트인 16개의 round-key K_i 유도
- 64 비트 (입력 key) but 각 8번째 비트는 앞 7 비트의 홀수 패리티 비트로 사용

  → parity bit : 실제 키 비트 X, 안전성 증대에 영향 X

- PC-1 : parity bit 제거

  → 56비트로 축소 (parity drop)

- PC-2 : 축약 전치

  → 56 bit (입력비트) 전치 후 → 48 bit

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

PC−2							
14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

03. DES (Data Encryption Standard)

❖ 키 스케줄 과정

① PC -1

: Parity bit (8 bit) 제거 → 56비트

② 56 비트 키는 C_0와 D_0로 반으로 나뉨

③ 각 28 비트는 순환 : round 별 규칙따라 왼쪽으로 순환 이동

: i = 1,2,9,16 (왼쪽으로 1 비트) & 나머지 round (왼쪽 2 비트)

③ 순환 위치의 수 ((4*1) + (12*2)) = 28 → $C_0 = C_{16}$, $D_0 = D_{16}$????

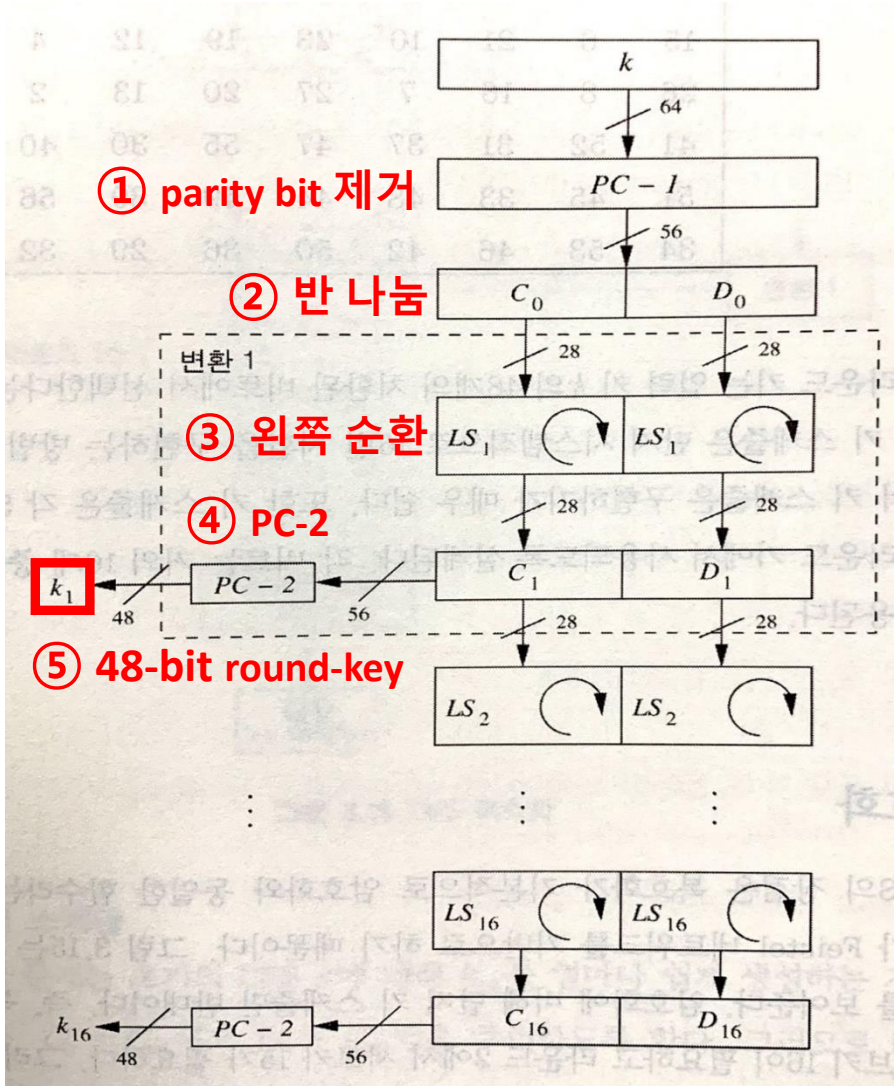
: 복호화 키스케줄링에 유용

④ PC-2

: 48 비트 round-key (K_i) 유도 위해 입력비트(C_i, D_i) 전치

→ 8개 제거 (56 → 48 비트)

⑤ K_i : 48 bit의 round-key 생성



03. DES (Data Encryption Standard)

❖ 안전성

Avalance Effect (쇄도효과) & Completeness (완비성) 이 높아 암호문에서 평문 추론 어려움

** Avalance Effect : 평문 또는 키 값의 작은 변화가 암호문에는 매우 큰 변화를 가져오는 성질

** Completeness : 암호문의 각 비트가 평문의 많은 비트들에 의존

But 소모적 키탐색(무작위 공격)에 취약

56 비트의 키 사용 → 키 공간 크기(2^{56})가 너무 작아 현대에는 안전하지 않음

→ 중요 정보의 암호화에는 사용할 수 없어짐

→ 단일 DES는 단기적 안전성 필요 시 or 암호화된 데이터 가치가 매우 낮은 분야에 사용

→ AES의 적용 전까지 3DES 사용

04. DES 예제

04. DES 예제

❖KEY

1. Key = HANSUNG : 8 bit 7개 : 56 bit

: 01110010 01100101 01111000 10000011 10000101 01111000 01110001

2. Parity bit 추가 : 8 bit 8개 : 64 bit

: 01110010 00110011 01011111 00010001 00111001 00101011 11100001 11100010

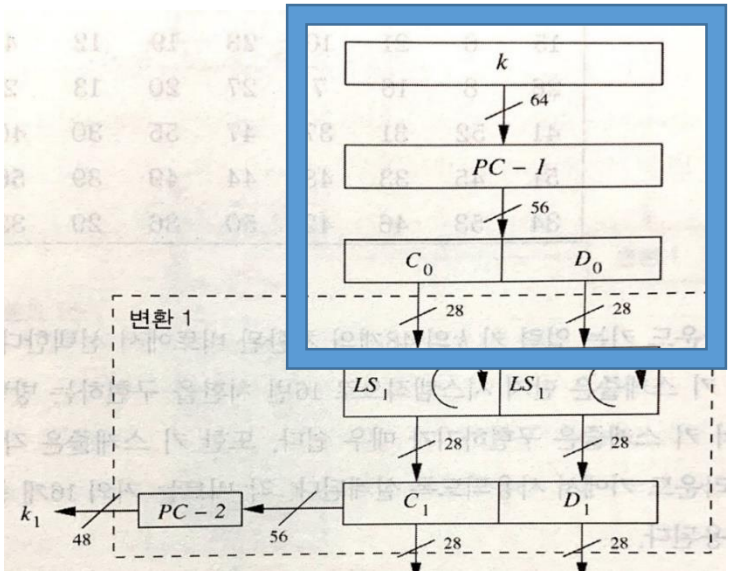
3. PC-1 적용

: 11000000 11000101 11110011 00011111 01110000 01000011 01001111 (56bit)

4. 반으로 나눔

: C_0 = 11000000 11000101 11110011 0001 (28bit)

: D_0 = 11110111 00000100 00110100 1111 (28bit)



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

3번 과정의 PC -1

04. DES 예제

5. Round 규칙 따라 왼쪽 shift (1round 이므로 1bit left shift)

: C_1: 10000001 10001011 11100110 0011

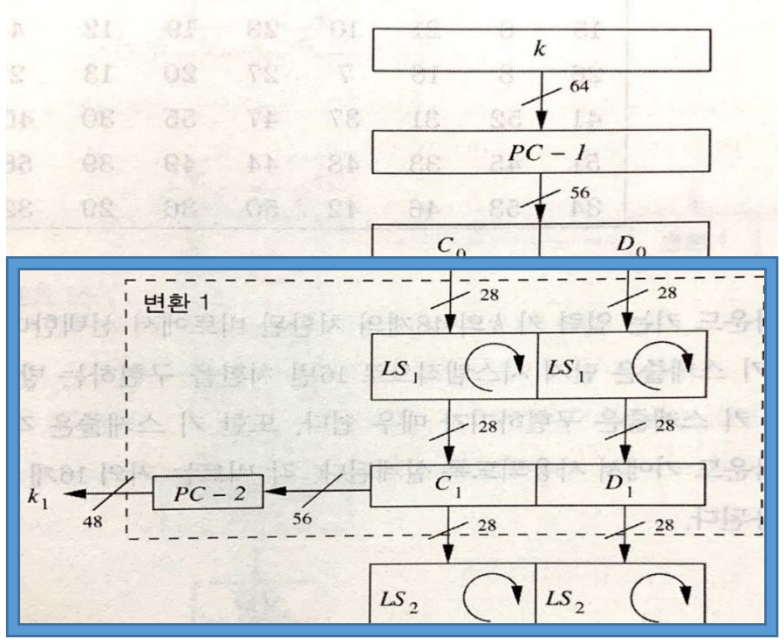
: D_1 : 11101110 00001000 01101001 1111

→ 다음 round-key는 이 C_1과 D_1으로 생성

6. PC-2 적용

: **K_1** = 01001001 10001100 01101010 11101110 00100101 11100010 (48 bit)

➢ 각 round의 5번 과정의 C_i과 D_i으로 5~6번 과정 반복하여 round-key 생성



PC−2							
14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

6번 과정의 PC -2

04. DES 예제

❖Encryption

1. 평문 : PLAINTEXT: 72bit → PLAINTEX : 64bit (block cipher)

: 01010000 01001100 01000001 01001001 01001110 01010100 01000101 01011000

2. Initial Permutation

: 11111111 10100001 01110010 01001100 00000000 00000000 10011010 00010000

3. 반으로 나눔 (각 32 bit)

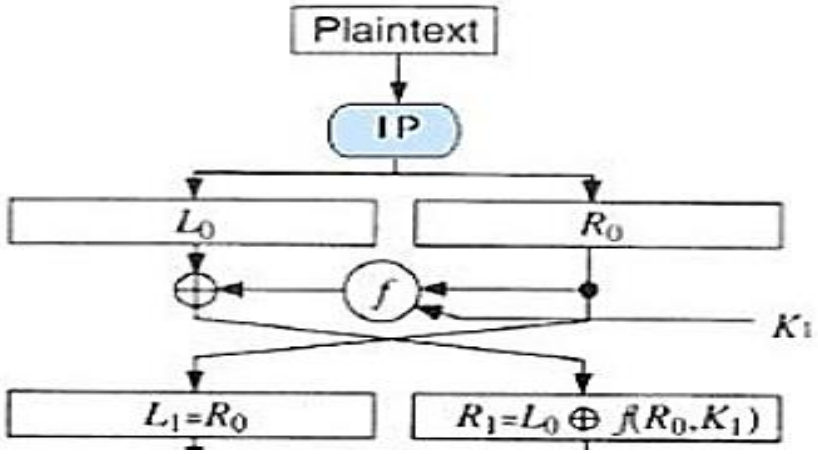
: L_0 = 11111111 10100001 01110010 01001100

: R_0 = 00000000 00000000 10011010 00010000

4. Expansion Permutation : E(R_0)

: 000000 000000 000000 000001 010011 110100 000010 100000 (48bit)

: subkey와 길이 같음



E					
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

4번 과정의 E-box (확장전치)

04. DES 예제

5. round-key와 확장된 R_0 XOR

: K_1 = 010010011000110001101010111011100010010111100010

: R_0 = 00000000000000000000001010011110100000010100000

: $K_1 \oplus R_0$ = 010010011000110001101011101000010110010101000010

6. 6bit 8개로 나눔

: 010010 011000 110001 101011 101000 010110 010101 000010

7. S-box : 1 & 6bit (행) , 2 &3&4&5 bit (열)

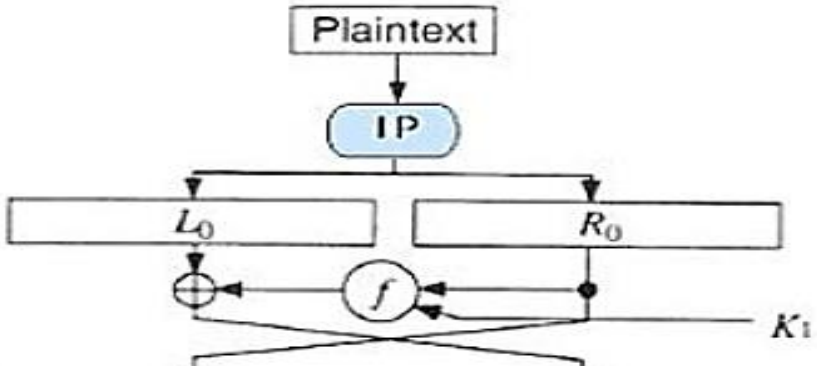
: $S_1(0,9)$ / $S_2(0,10)$ / $S_3(3,8)$ / $S_4(3,5)$ / $S_5(2,4)$ / $S_6(0,11)$ / $S_7(1,10)$ / $S_8(0,1)$ → 10 12 4 1 10 4 5 2

: 결과 값을 binary로 바꿈 → 1010 1100 0100 0001 1010 0100 0101 0010 (32bit)

8. P-box

: 10000011 10011011 00000010 10011000 (32bit)

: L_0과 길이 동일



	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0yyyy1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
1yyyy0	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
1yyyy1	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S-box 1

	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
0yyyy1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
1yyyy0	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
1yyyy1	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S-box 2

																8
																1
																7
																12
																1111x
																15
																9
																4
																14

P

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

04. DES 예제

9. $L_0 \oplus f(R_0,k_1)$

: L_0 = 11111111 10100001 01110010 01001100

: $f(R_0,k_1)$ = 10000011 10011011 00000010 10011000

: $L_0 \oplus f(R_0,k_1)$ = 01111100 00111010 01110000 11010100 (32bit)

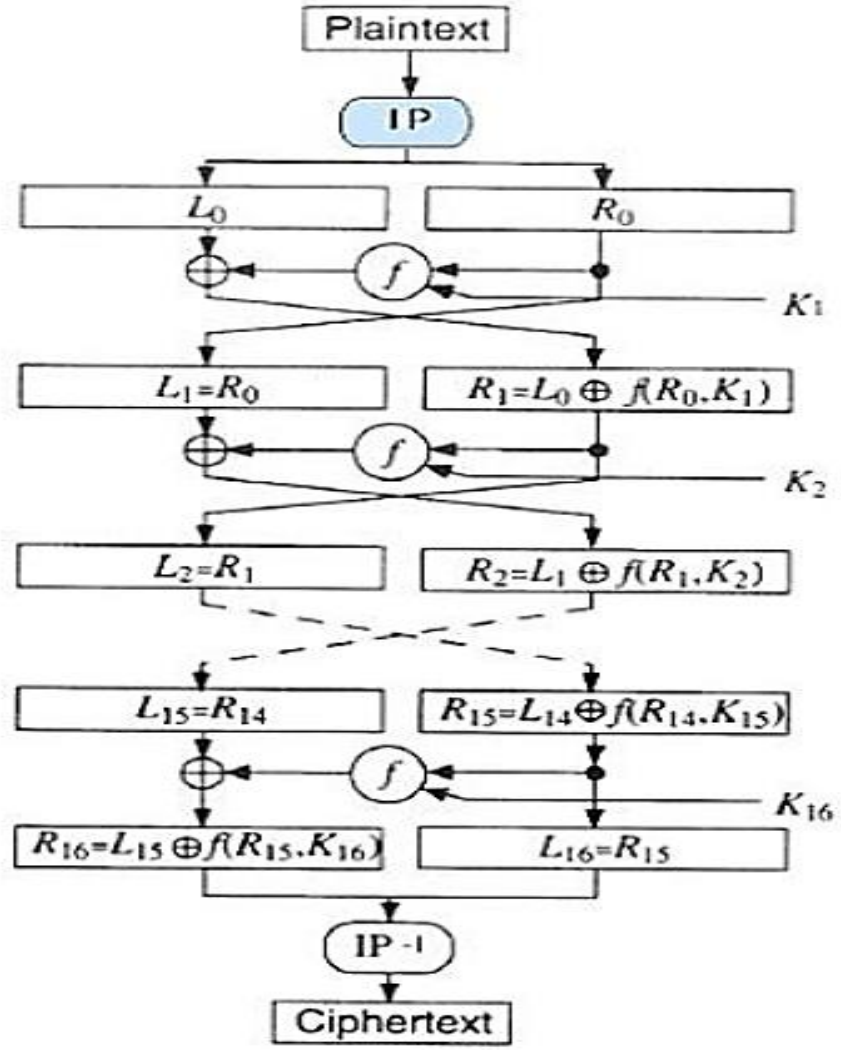
10. L, R 교환

: $L_1 = R_0$ = 00000000 00000000 10011010 00010000

: $R_1 = L_0 \oplus f(R_0,k_1)$ = 01111100 00111010 01110000 11010100

➢ 4~10 의 과정을 16-round까지 반복 → L_{16}, R_{16} 구한 후, 자리 바꾸고 IP^{-1} 적용

- 끝 -



Q & A