제 3 장 블록암호 와 DES

❑ 목 차

- ❖ 단순 **DES**
- ❖ 블록 암호 기법
- ❖ **DES**
- ❖ 블록 암호의 설계 원리
- ❖ 블록 암호의 운용 모드

암호학(cryptology)

암호시스템 3가지 영역

❖평문을 암호화하기 위한 연산자의 유형

- 전치(轉置, Transposition) : 평문의 각 원소를 재배열
- 치환(置換, Substitution) : 평문의 각 원소를 다른 원소로 사상

❖사용된 키의 수

- 관용키(conventional key) : single-key, symmetric, secret-key; 송수신자가 같은 키를 사용
- 공개키(public key) : two-key, asymmetric, public-key; 송수신자가 다른 키를 사용

❖평문 처리 방법

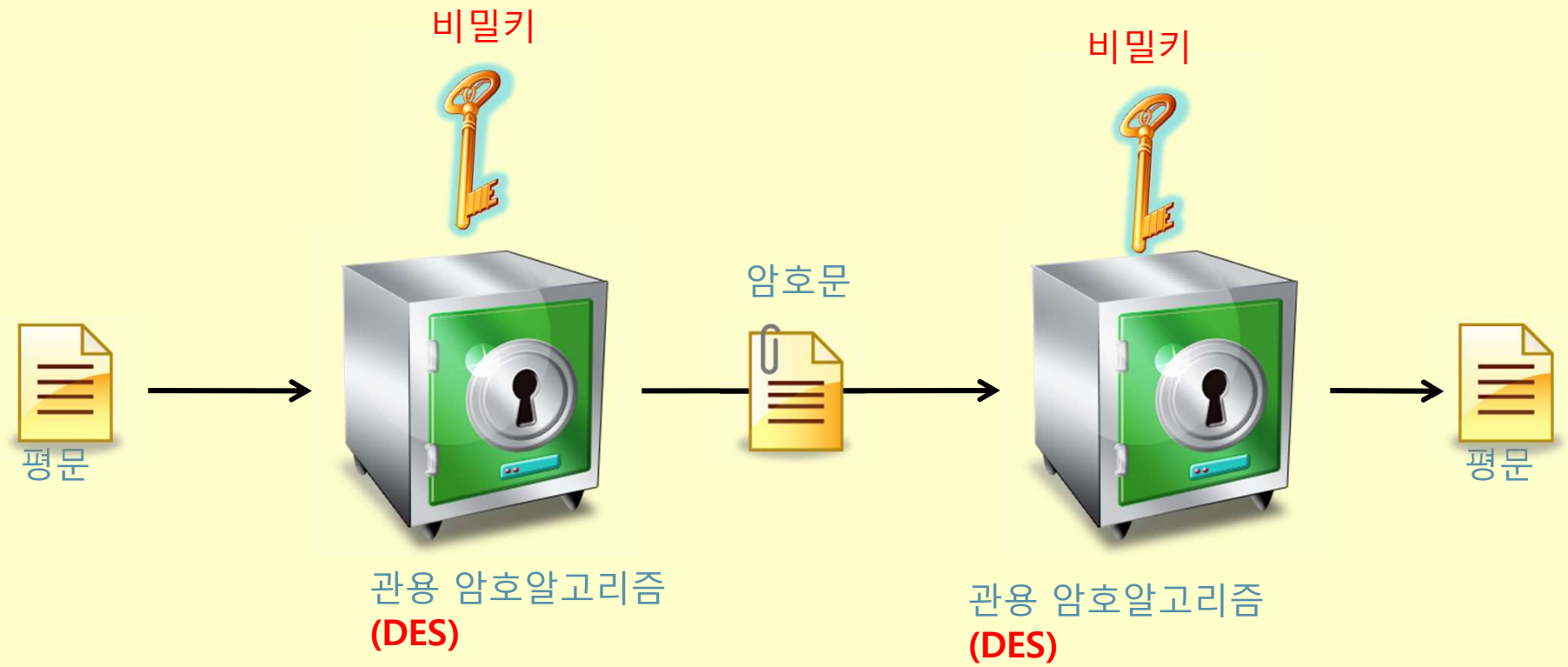
- 블록 암호화 (Block cipher) : 연산을 블록 단위로 처리
- 스트림 암호화 (Stream cipher) : 입력을 연속적으로 처리

블록 암호 기법

- 스트림 암호와 블록 암호 기법
  - 스트림 암호
    - 한번에 1비트 혹은 1 바이트
    - Vigenere 암호, Vernam 암호
  - 블록 암호
    - 평문 블록 전체(64비트 – 전형적)
    - 다양한 작동모드 사용
    - 스트림 방식에 비해 응용 범위 넓음
    - 대부분 네트워크 기반 관용 암호 방식에 사용

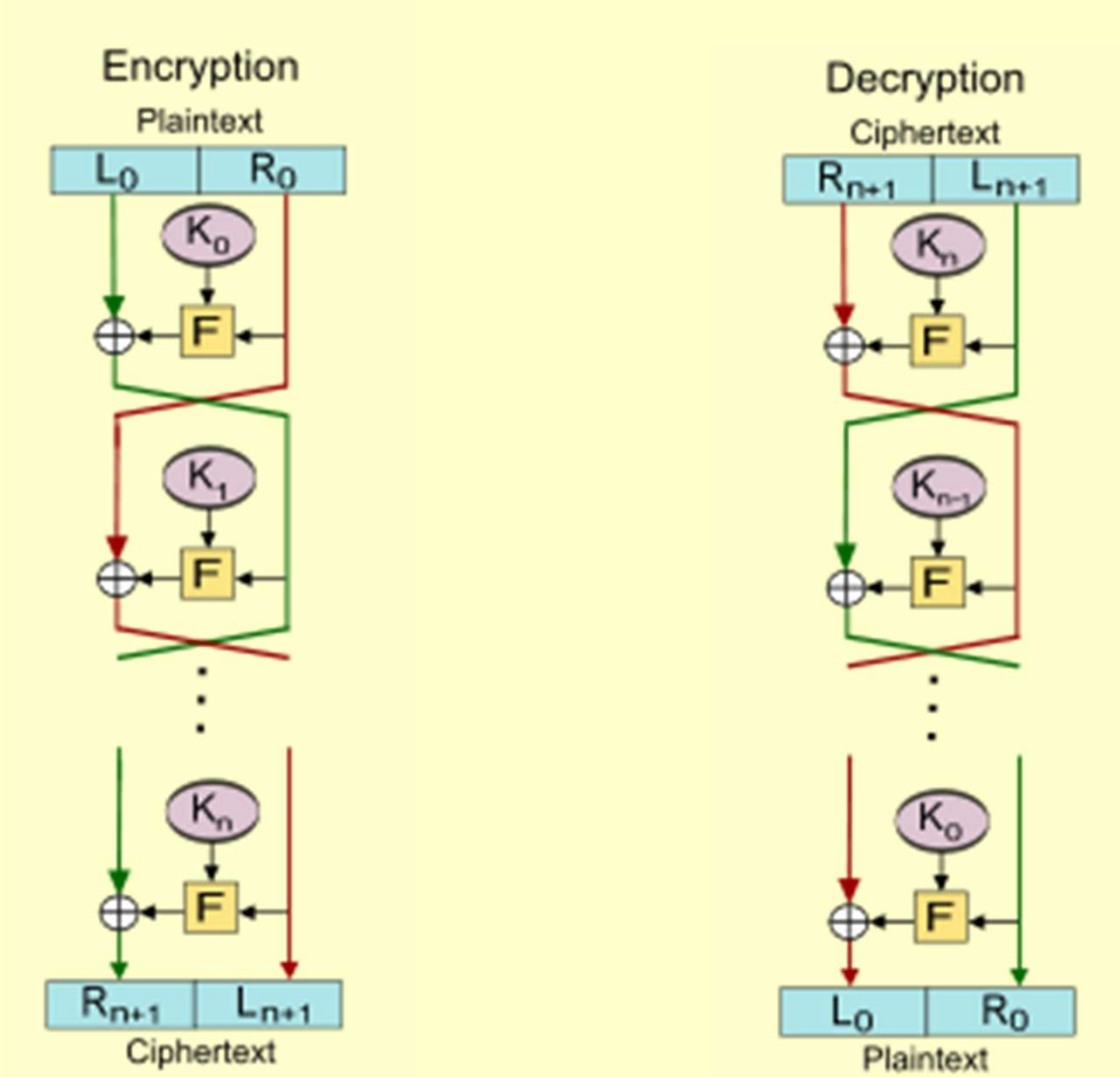
❑ 관용 암호 방식 (conventional key cryptosystem)

- ➢ 송수신자가 **같은 키**를 사용
- ➢ single-key, symmetric-key, secret-key



Feistel 암호

- 두 개 이상의 기본 암호 연속적 수행(치환, 순열(전치) 번갈아 수행)



Feistel 암호

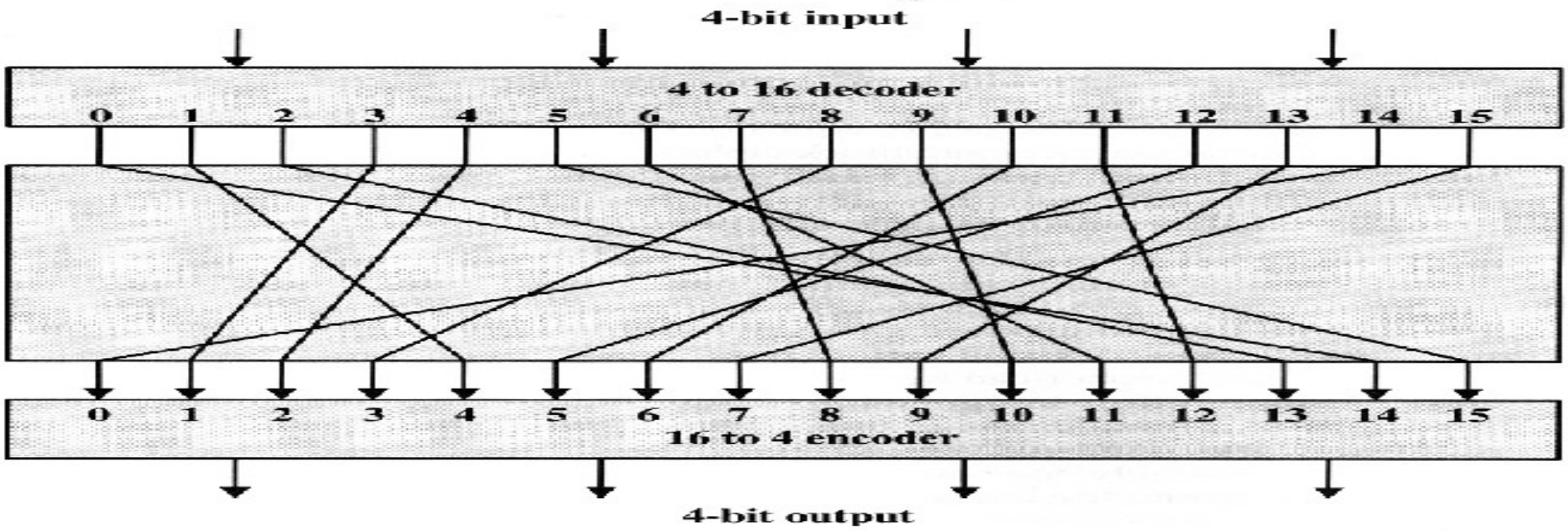
- Horst Feistel (January 30, 1915–November 14, 1990) was a cryptographer who worked on the design of ciphers at IBM, initiating research that would culminate in the development of the Data Encryption Standard (DES) in the 1970s.

Feistel 암호

- 확산과 혼돈
- 매우 이상적인 암호는 암호문에 대한 모든 통계적 정보가 사용된 키와 독립적이어야 한다.
  - Claude Shannon 소개
  - 통계적 분석에 기초한 암호 해독 방지
- 확산(diffusion)
  - 평문의 통계적 구조가 암호문에 광범위하게 분산(평문과 암호문 관계 복잡)
  - 각 평문 숫자가 다수의 암호문 숫자 값에 영향
- 혼돈(confusion)
  - 암호문의 통계적 구조와 암호 키 값 사이의 관계 복잡
  - 주키를 이용한 암호문 생성 방법 복잡(주키 추론 어려움)

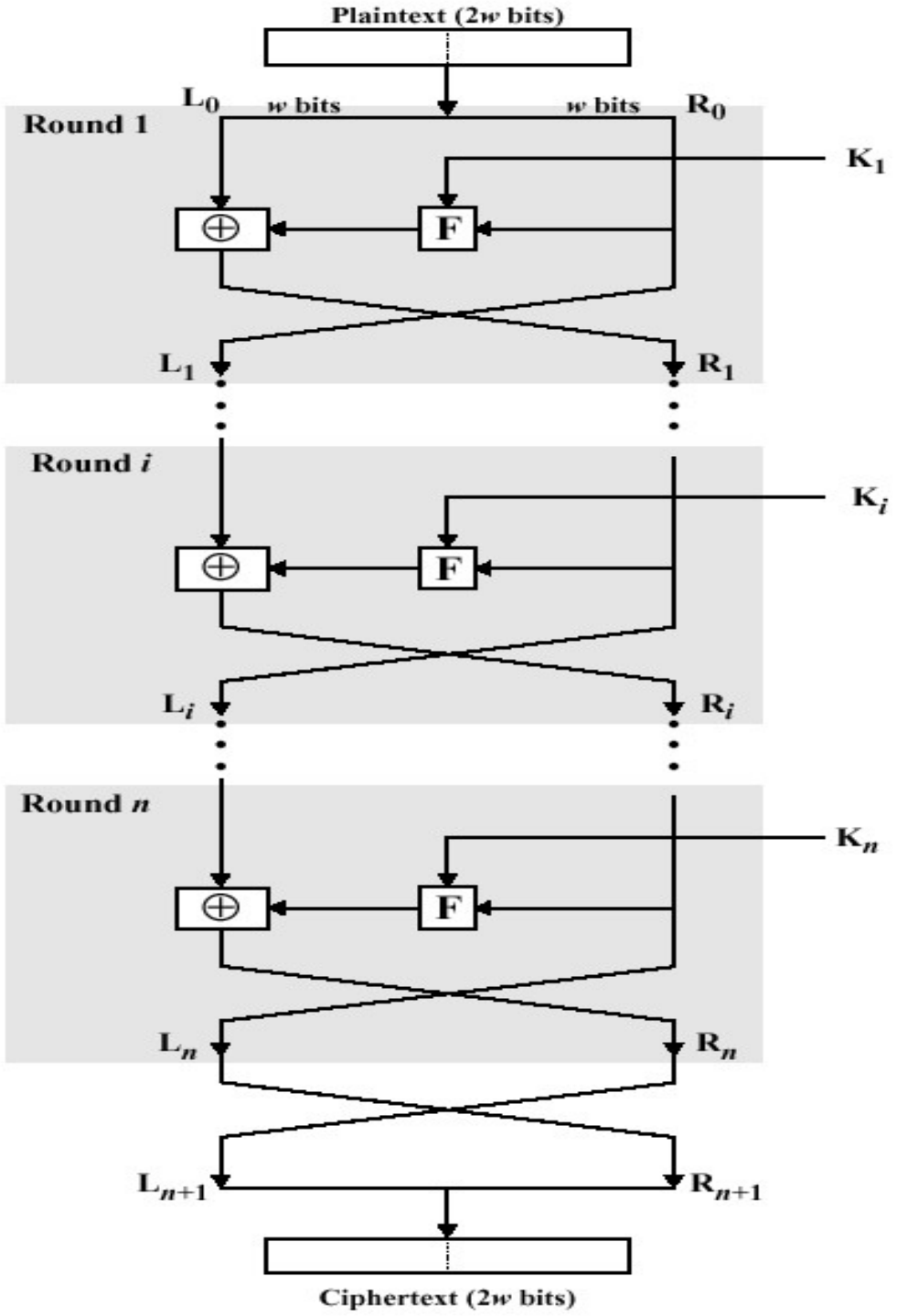
Feistel 암호 구조

- n비트 블록처리: n비트 평문을 입력으로 n비트 암호문 출력
  - 역으로 n 비트 암호문 입력에 대해 n비트 평문 출력(역의 성립: reversible, 비특이형: nonsingular)
  - 2^n 가지의 서로 다른 블록 존재 가능
- 일반적인 n 비트-n 비트 블록치환(n=4 인 경우)
  - 4비트 입력으로 16개 값 중 하나 선택하고 , 내부 치환에 의하여 16개 출력 값 중 하나 대응하여 4비트 출력


4-bit input
4 to 16 decoder
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 to 4 encoder
4-bit output

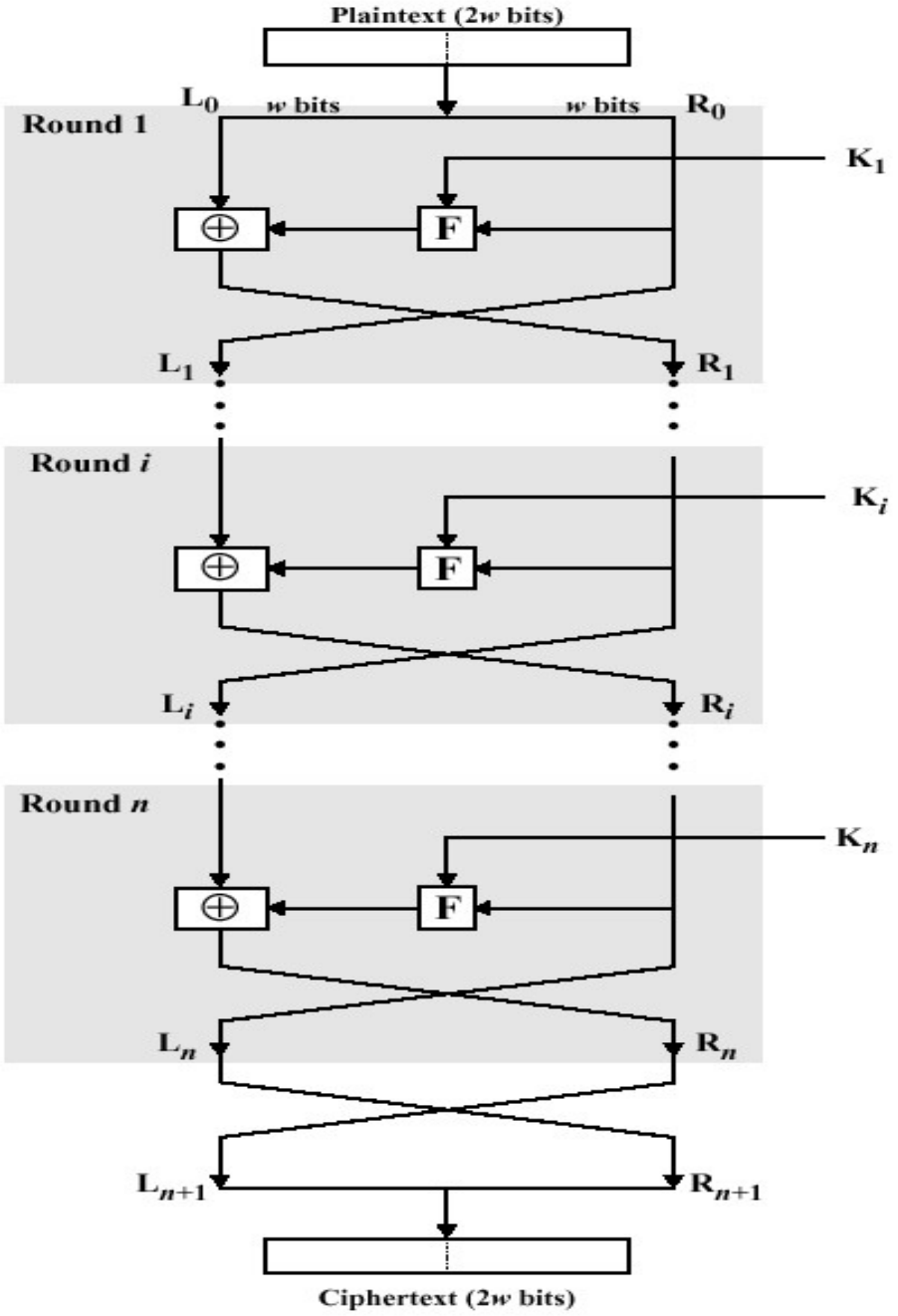
Feistel 암호 구조

- 처리구조
  - 길이 2w 비트인 평문 블록(L_0, R_0) 분할 처리
  - K로부터 유도된 n개의 키(K_i) 사용
  - n회의 동일한 반복 구조 실행
- 하나의 반복 구조
  - 오른 쪽 반 R0에 반복 함수 F 적용
    - 반복 서브키 K_i 적용($K_i \neq K_j$)
  - 왼쪽 반 L0와 XOR(치환 작용)
  - 좌우 양쪽 결과를 교환 (순열, 전치 작용)



Feistel 암호 구조

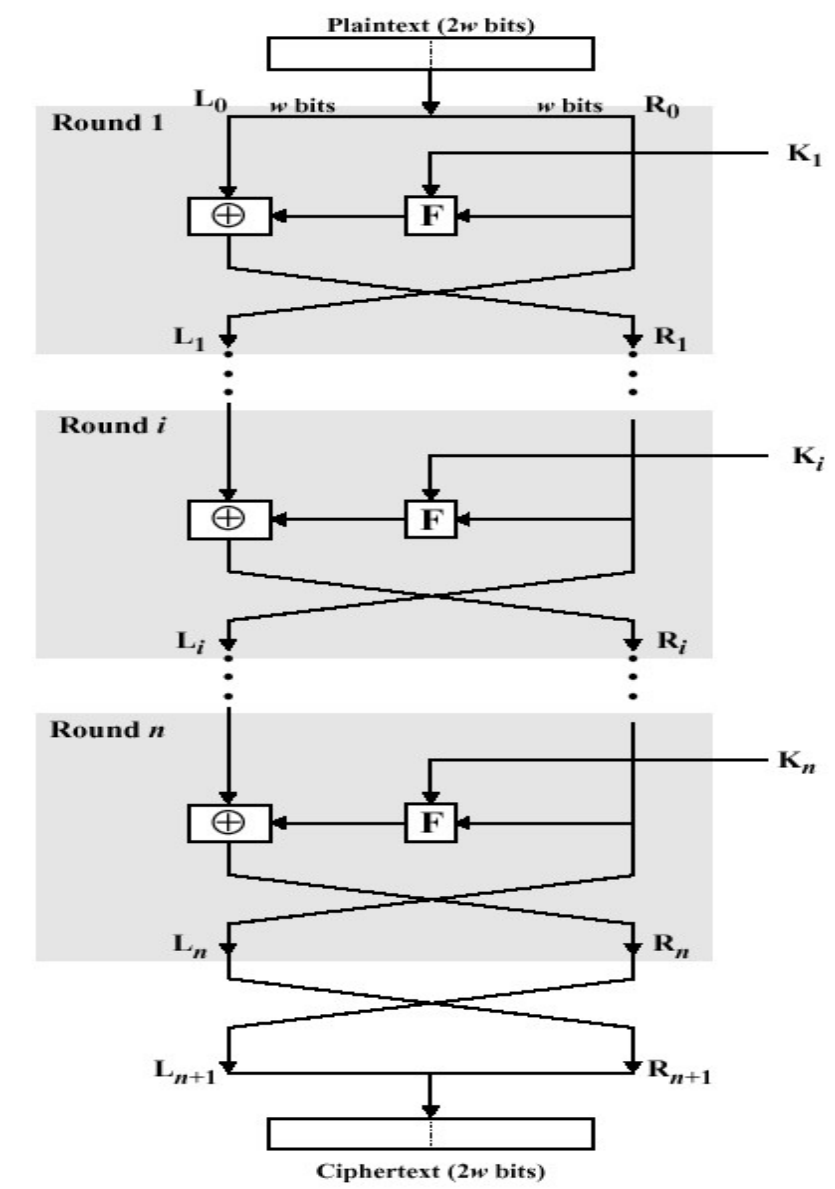
- 암호 알고리즘의 n번의 과정 :
  - $LE_1 = RE_0$
  - $RE_1 = LE_0 \oplus F(RE_0, K_1)$
  - $LE_2 = RE_1$
  - $RE_2 = LE_1 \oplus F(RE_1, K_2)$
  - $LE_i = RE_{i-1}$
  - $RE_i = LE_{i-1} \oplus F(RE_{i-1}, K_i)$
- n번의 반복과정
- 암호화 마지막 반복과정
  - $LE_{16} = RE_{15}$
  - $RE_{16} = LE_{15} \oplus F(RE_{15}, K_{16})$



Feistel 암호 구조

- 암호 알고리즘의 i 번째 반복 과정은
  - $LE_i = RE_{i-1}$
  - $RE_i = LE_{i-1} \oplus F(RE_{i-1}, Ki)$

- 역으로 정리하면
  - $RE_{i-1} = LE_i$
  - $LE_{i-1} = RE_i \oplus F(RE_{i-1}, K_i)$
    $= RE_i \oplus F(LE_i, K_i)$
  - ➔복호화는 (i-1) 번째 복호 결과를 위하여 (i)번째 입력 형식을 취함
  - $RE_1 = LE_2$
  - $LE_1 = RE_2 \oplus F(RE_1, K_2)$
    $= RE_2 \oplus F(LE_2, K_2)$



Feistel 암호 구조

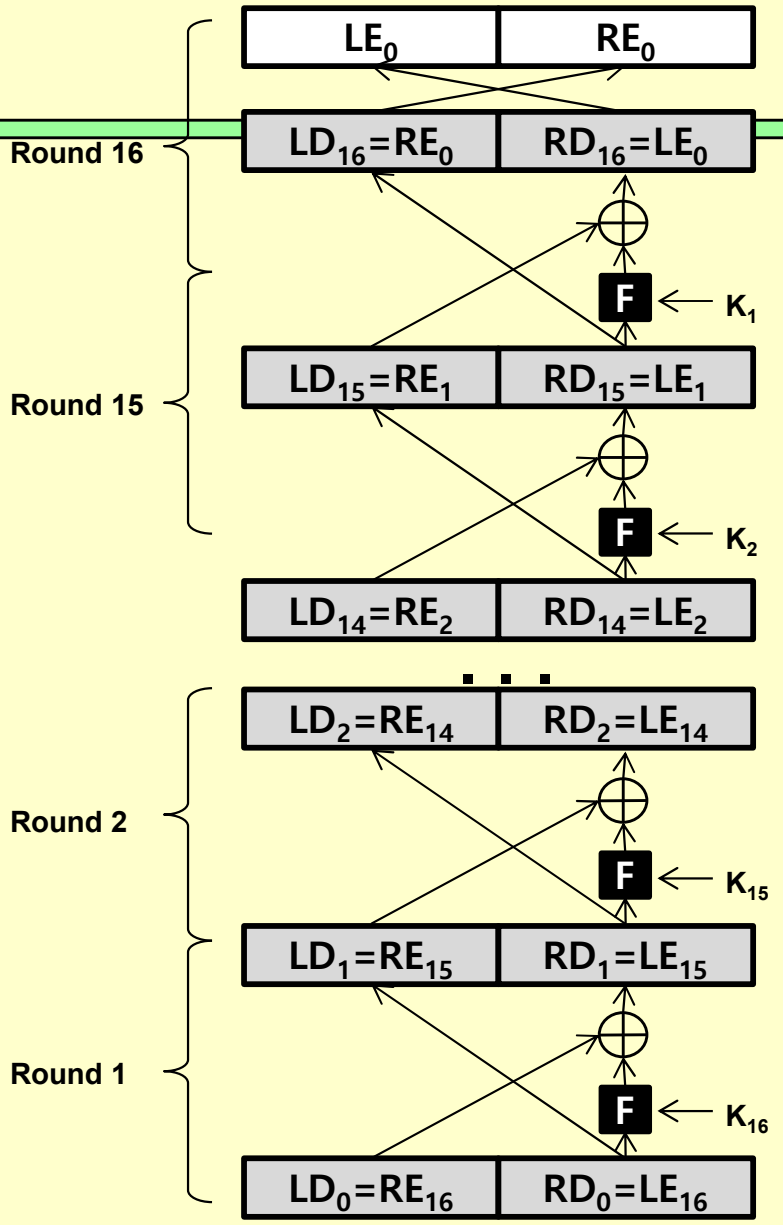
- 암호화 마지막 반복
  - $LE_{16} = RE_{15}$
  - $RE_{16} = LE_{15} \oplus F(RE_{15}, K_{16})$
- 복호화 과정
- 암호화과정의 역 순서 처리
- 첫 반복
  - $LD_1 = RD_0 = LE_{16} = RE_{15}$
  - $RD_1 = LD_0 \oplus F(RD_0, K_{16})$

    $= LD_0 \oplus F(LE_{16}, K_{16})$

    $= RE_{16} \oplus F(RE_{15}, K_{16})$

    $= [LE_{15} \oplus F(RE_{15}, K_{16})] \oplus F(RE_{15}, K_{16})$

    $= LE_{15}$



대칭 블록 암호 구현

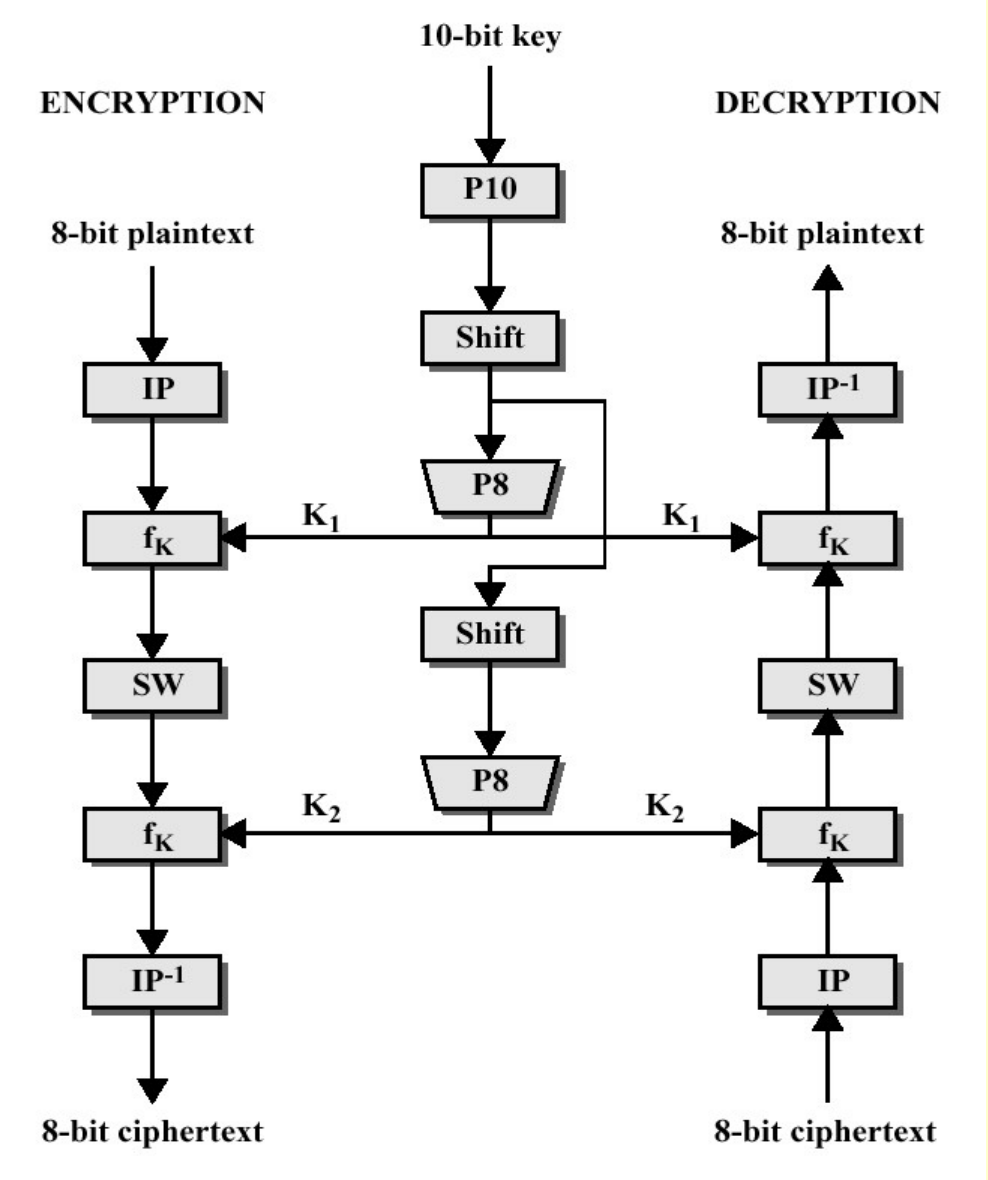
- **블록크기(Block size)** : 블록 길이가 크다는 것은 더 강한 보안을 의미하지만(다른 조건이 같다는 가정 하에) 길이가 길면 암 · 복호화 속도는 떨어진다
- **키 길이(Key size)** : 키 길이가 길다는 것은 더 강한 보안을 의미하지만 암 · 복호화 속도는 떨어진다.
- **라운드 수(Number of rounds)** : 단일 과정으로는 보안이 부족하지만 라운드 수를 증가시켜 여러 번 수행하면 보안을 강화할 수 있다는 것이 Feistel 암호의 핵심이다.
- **서브키 생성 알고리즘** : 이 알고리즘이 복잡하면 복잡할수록 암호해독이 어려워진다.
- **반복 함수** : 이 함수 역시 복잡하면 복잡할수록 암호해독이 어려워진다.

DES (Data Encryption Standard)

- 전용 암호시스템의 필요성 대두
  - 타 그룹간의 통신에 불리 : 데이터 암호화 표준이 필요
- 미연방 정부에 의해 공개 암호 표준화 작업 진행
- 1977년 미 상무성의 국립 표준국(National Bureau of Standards)에서 연방 정보처리 표준 채택 46(FIPS PUB46)
- 64비트 평문, 56비트 키를 사용
- 치환과 전치 혼합방법, 블럭 암호방식, 관용암호방식

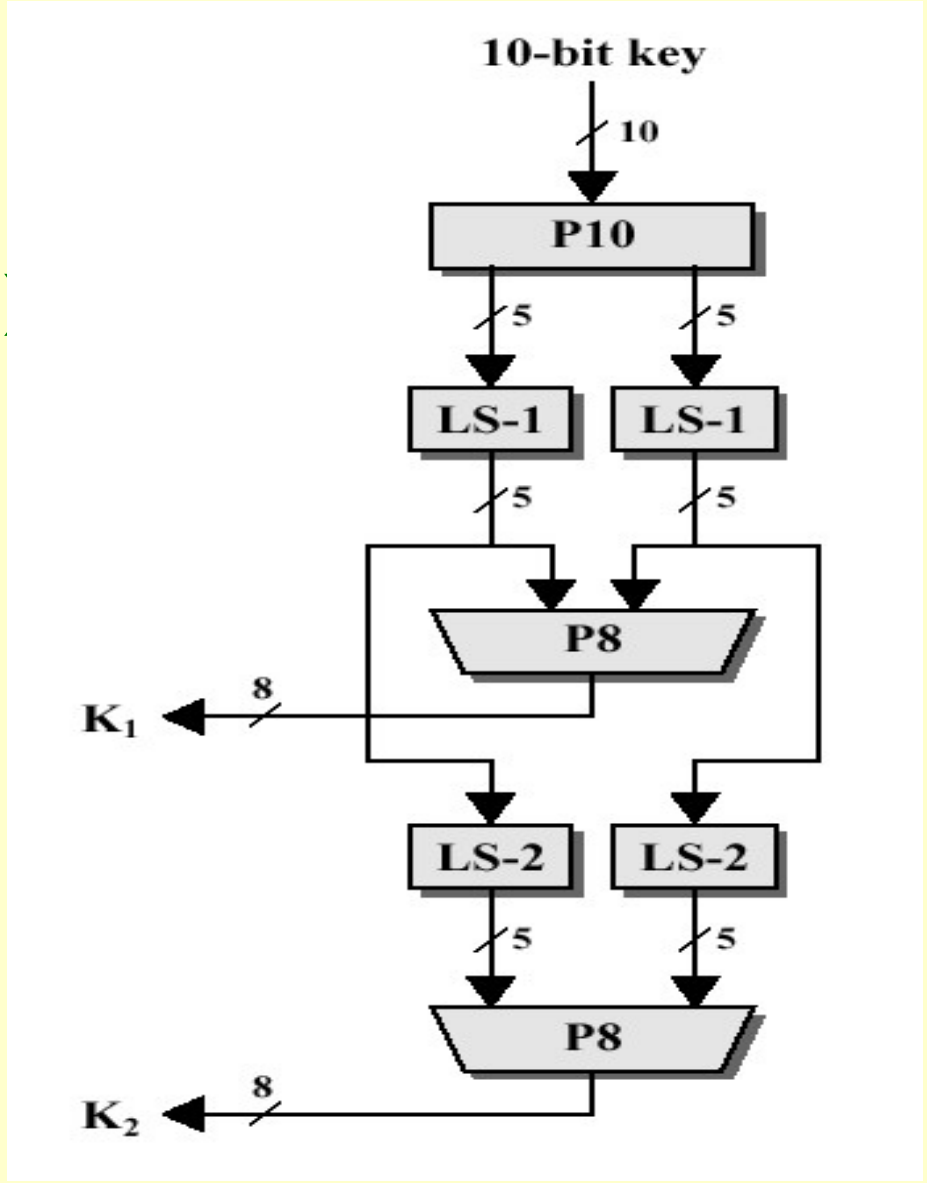
단순 DES[Simplified Data Encryption Standard]

- 알고리즘
  - 8비트 평문, 10비트 키
  - 8비트 암호문 생성
    - 초기순열(IP)
    - 순열, 치환 이용 f_k
    - 순열 함수 SW
    - 역 순열(IP^{-1})
- 암호문
  - $IP^{-1}(f_{k_2}(SW(f_{k_1}(IP(평문)))))$
- 복호문
  - $IP^{-1}(f_{k_1}(SW(f_{k_2}(IP(암호문)))))$



S-DES를 위한 키 생성

- K_1 = P8(Shift(P10(key)))
- K_2 = P8(Shift(Shift(P10(key))))



S-DES 키의 생성(1/4)

- K_1 = P8(Shift(*P10(key)*))
- 10 비트 키 = (k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10})
- P10 = (k_3, k_5, k_2, k_7, k_4, k_{10}, k_1, k_9, k_8, k_6)

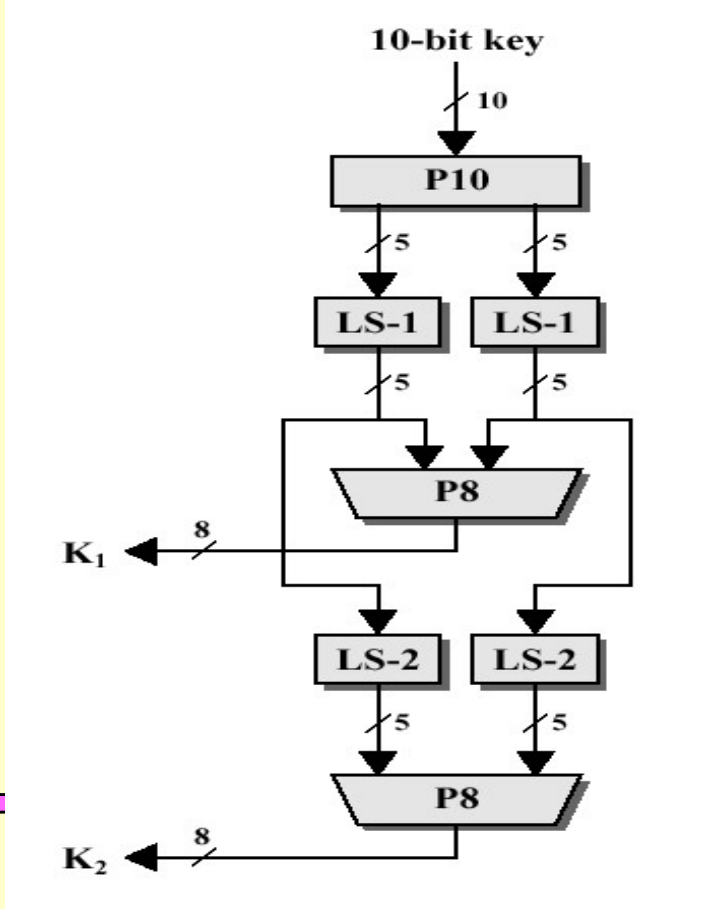
1	2	3	4	5	6	7	8	9	10

P10									
3	5	2	7	4	10	1	9	8	6

❖예)

❖10 비트 *key* = (1 0 1 0 0 0 0 0 1 0)

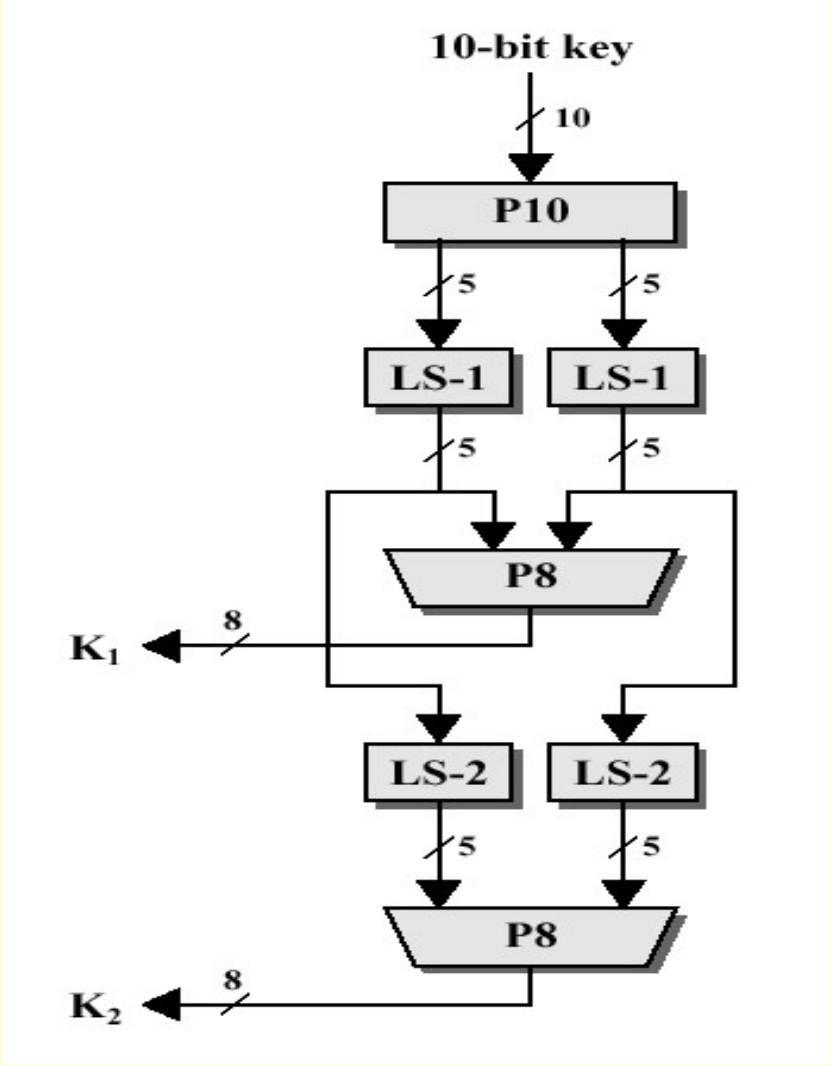
❖*P10(key)* = (1 0 0 0 0 0 1 1 0 0)



(2/4)

- K_1 = P8(*Shift(P10(key))*)
- (*Shift(P10(key))*): LS-1[키의1st 5비트] & LS-1[키의 2nd 5비트]
  - 첫 번째 다섯 비트와 두 번째 다섯 비트 좌로 순환 이동
  - 1비트 좌측 순환이동

- P10 =(k_3, k_5, k_2, k_7, k_4, k_{10}, k_1, k_9, k_8, k_6)
- Shift =(k_5, k_2, k_7, k_4, k_3, k_1, k_9, k_8, k_6, k_{10})



(3/4)

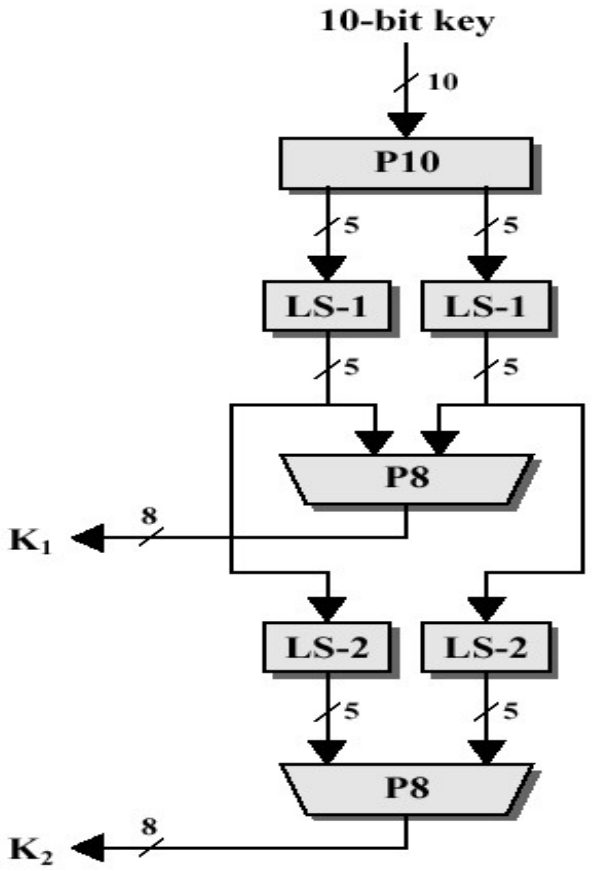
- 예제)

  1 2 3 4 5 6 7 8 9 0

  - P10 = (1 0 0 0 0 0 1 1 0 0)
  - LS-1 = (0 0 0 0 1 1 1 0 0 0)

- K_1 = *P8(Shift(P10(key)))* = P8(LS-1)
- P8(LS-1) = P8(0 0 0 0 1 1 1 0 0 0)
  - 10 비트에서 8비트 선택 치환

P8
6 3 7 4 8 5 10 9



- K_1 = (1 0 1 0 0 1 0 0)

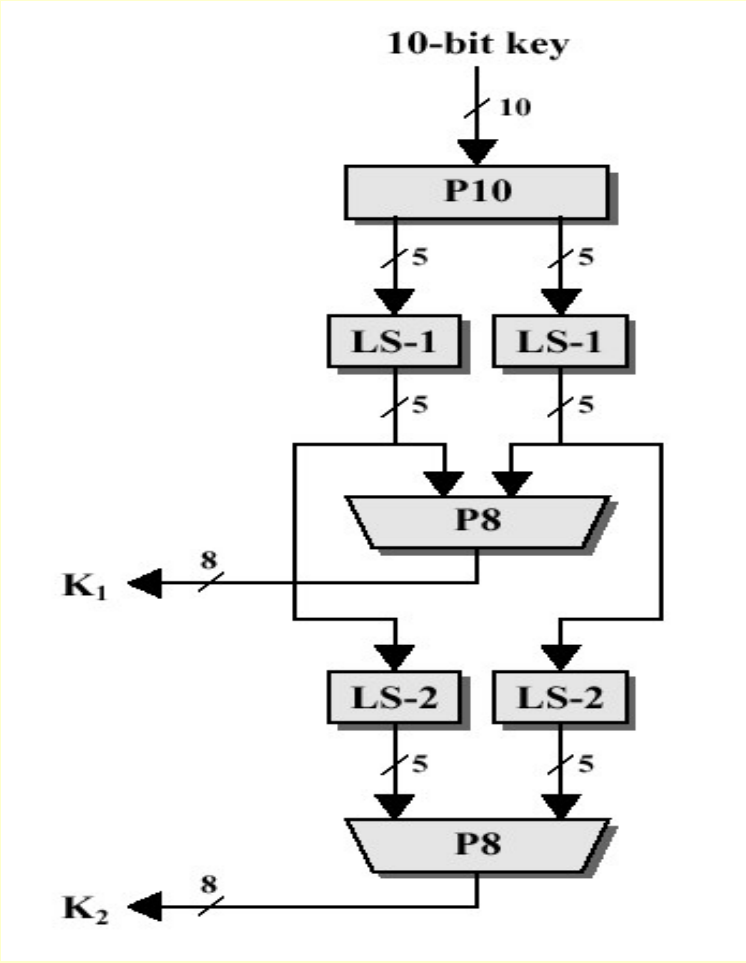
(4/4)

- K_2 = *P8(Shift(Shift(P10(key))))* = *P8(Shift(LS-1))* = P8(LS-2)
- *LS-2 = Shift(LS-1)*
  - LS-1의 결과에 2비트 좌측 순환 이동
  - LS-1 : (0 0 0 0 1 1 1 0 0 0)
- K_2 = P8(LS-2)

  = P8(0 0 1 0 0 0 0 0 1 1)

P8
6 3 7 4 8 5 10 9

- K_2 = (0 1 0 0 0 0 1 1)



Quiz 1

- 키의 값이 **1 1 0 0 1 0 1 0 0 1** 인 경우 **S-DES**에서의 **2**개의 세션 키**(K_1, K_2)** 값은 얼마인가**??**

P10									
3	5	2	7	4	10	1	9	8	6

❖10 비트 *key* = (1 1 0 0 1 0 1 0 0 1)
1 2 3 4 5 6 7 8 9 0

❖*P10(key)* = ( )

❖LS-1 = ( )

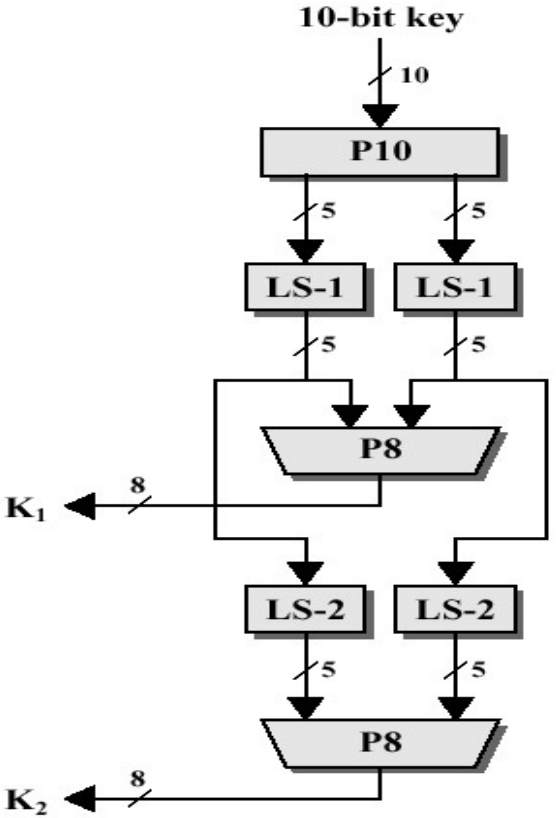
❖P8(LS-1)

❖ K1 = ( )

❖ LS-2 = ( )

❖ P8(LS-2)

❖K2 = ( )

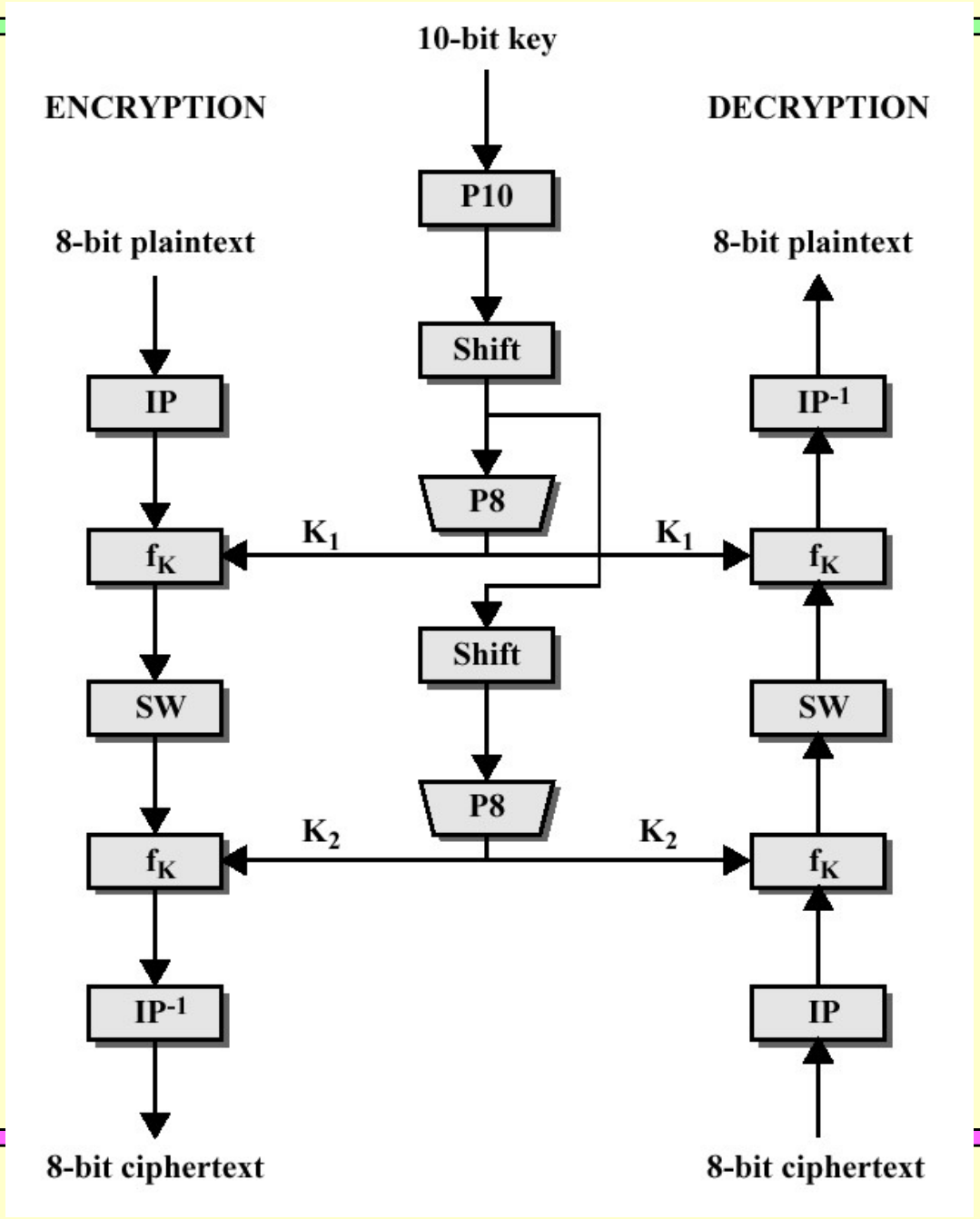


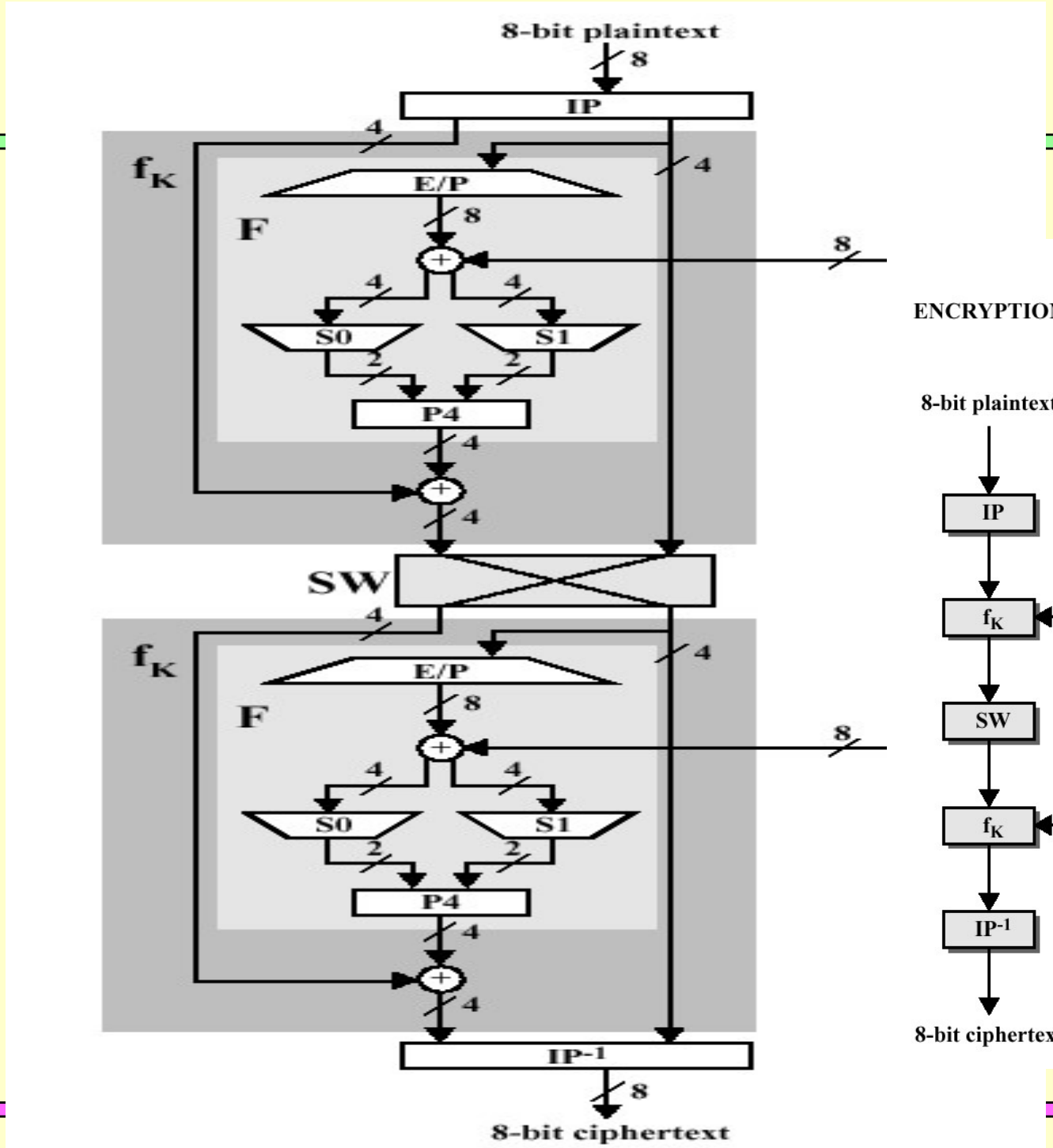
P8							
6	3	7	4	8	5	10	9

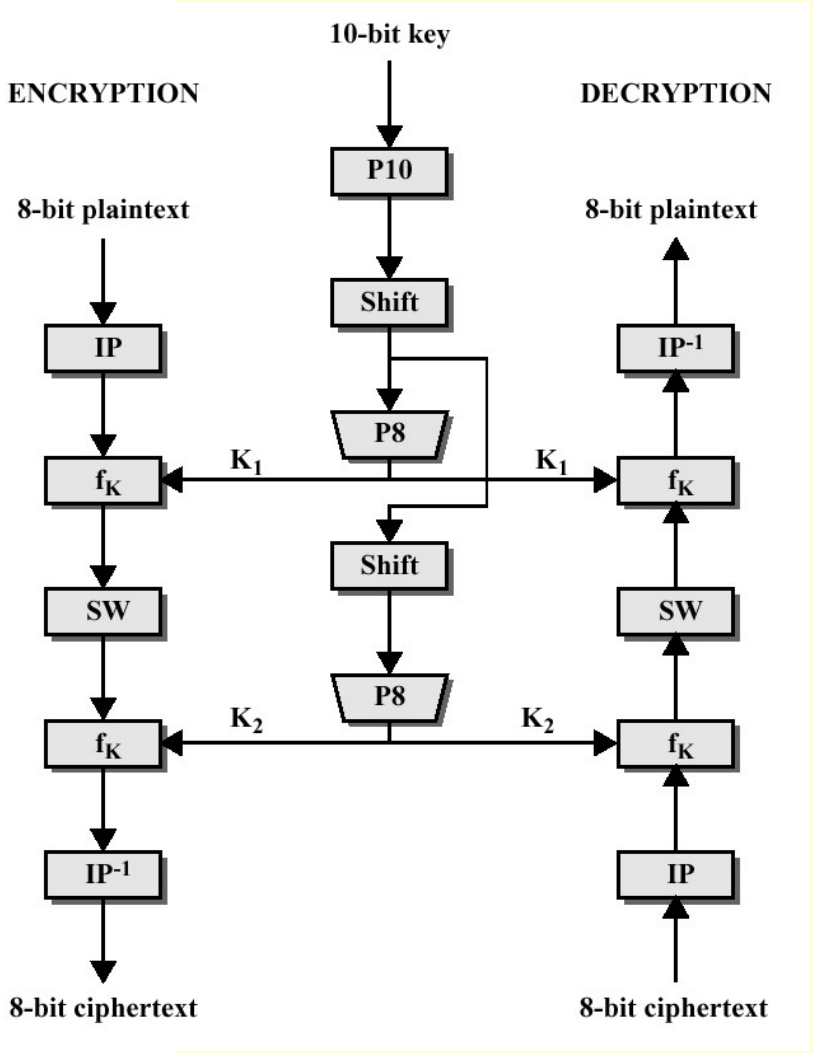
Quiz 2

- 키의 값이 **1 0 1 0 1 0 1 1 0 1** 인 경우 **S-DES**에서의 **2**개의 세션 키**(K_1, K_2)** 값은 얼마인가**??**

S-DES 암호 알고리즘


10-bit key
ENCRYPTION
DECRYPTION
P10
8-bit plaintext
8-bit plaintext
Shift
IP
IP-1
P8
K1
K1
fK
fK
Shift
SW
SW
P8
K2
K2
fK
fK
IP-1
IP
8-bit ciphertext
8-bit ciphertext

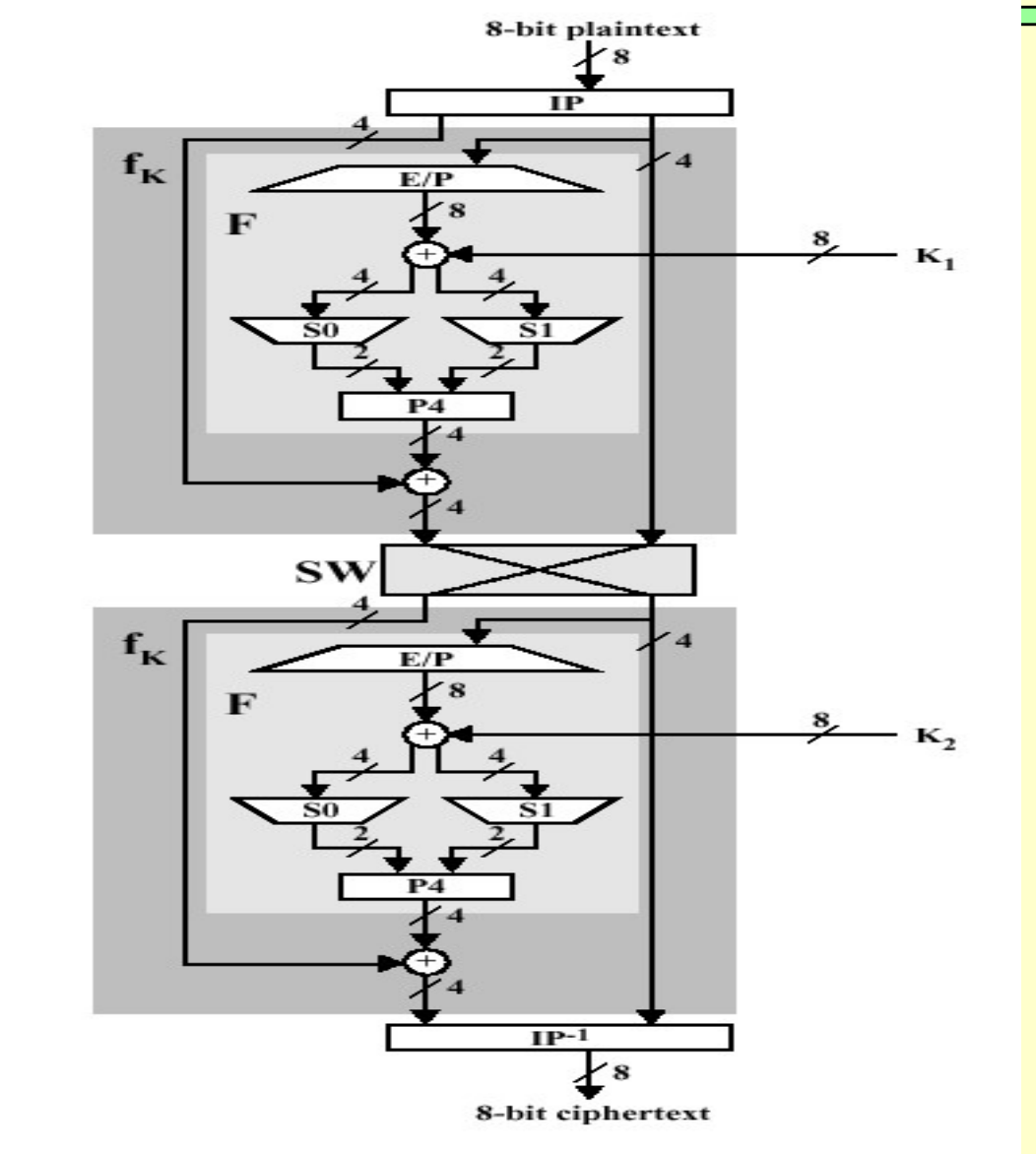

8-bit plaintext
8
IP
4
4
f_K
E/P
F
8
8
4
4
S0
S1
2
2
P4
4
4
SW
4
f_K
E/P
F
8
8
4
4
S0
S1
2
2
P4
4
4
IP-1
8
8-bit ciphertext


10-bit key
ENCRYPTION
DECRYPTION
P10
8-bit plaintext
8-bit plaintext
Shift
IP
IP-1
P8
K_1
K_1
f_K
f_K
Shift
SW
SW
P8
K_2
K_2
f_K
f_K
IP-1
IP
8-bit ciphertext
8-bit ciphertext

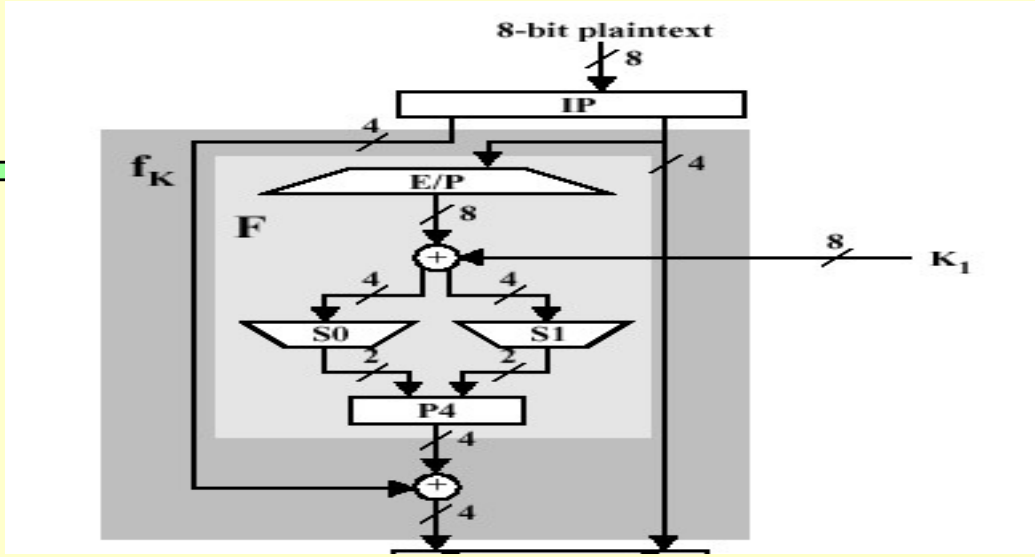
함수 f_k(1/5)

- 순열, 치환 함수 조합
- L(Left)
  - 왼쪽 4비트
- R(Right)
  - 오른쪽 4비트
- f_k(L, R) = (L ⊕ F(R, SK), R)
- F 함수(확장 순열)

E/P							
4	1	2	3	2	3	4	1



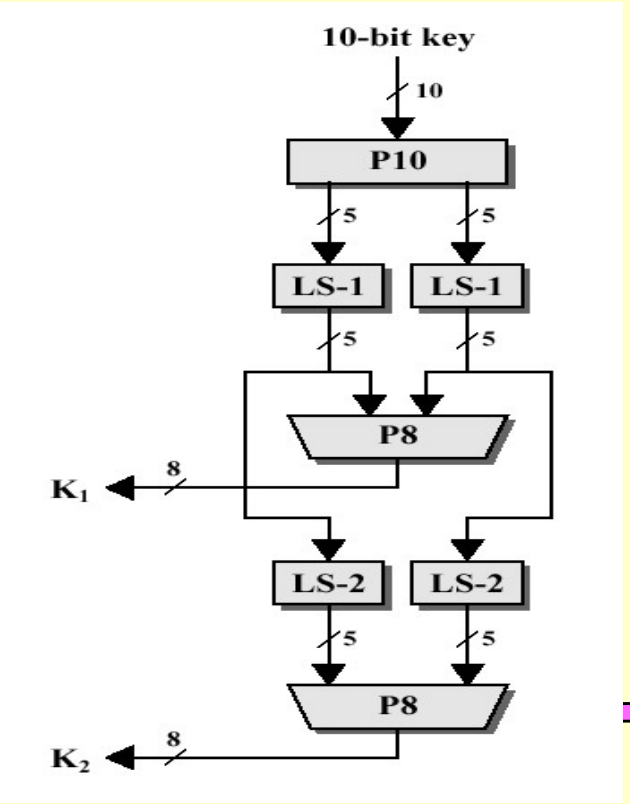
(2/5)



- 확장 순열(E/P); 4비트➔ 8비트

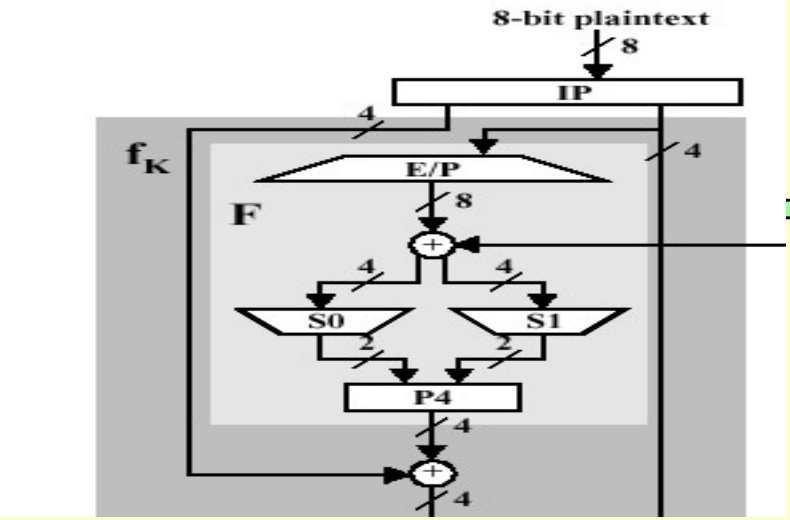
E/P
4 1 2 3 2 3 4 1

  - R= (0 1 1 0)
  - 결과 = (0 0 1 1 1 1 0 0)



- XOR
  - 서브키 K_1(8bit)⊕ 확장 순열 결과(8bit)
  - $K_1 = (k_{11}, k_{12}, k_{13}, k_{14}, k_{15}, k_{16}, k_{17}, k_{18})$
  - 결과 = $(o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8)$

(3/5)



S-Box

- XOR결과 = (o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8)
- 4비트 입력, 2비트 출력
- 입력 = (o_1, o_2, o_3, o_4, o_5, o_6, o_7, o_8)
- S0 : 1,4 요소 ➔ 행, 2,3 요소 ➔ 열
- S1 : 5,8 요소 ➔ 행, 6,7 요소 ➔ 열

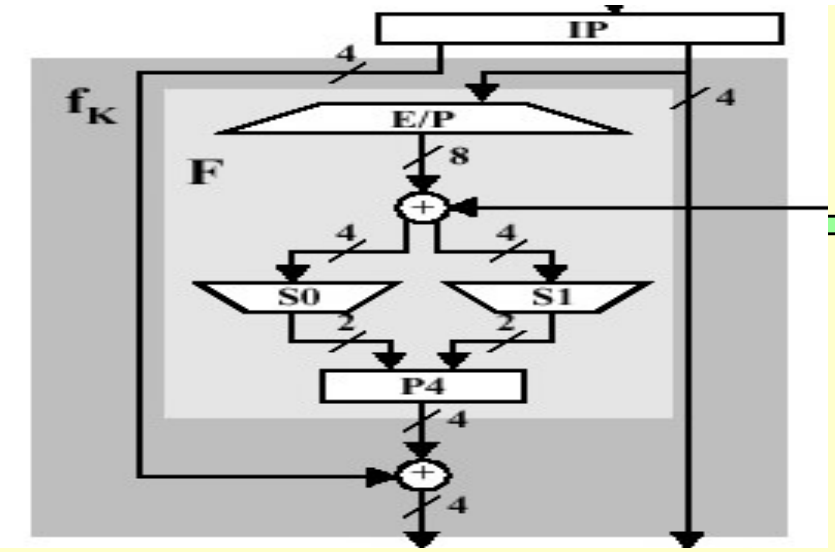
S0 =

	0	1	2	3
0	1	0	3	2
1	3	2	1	0
2	0	2	1	3
3	3	1	3	2

S1 =

	0	1	2	3
0	0	1	2	3
1	2	0	1	3
2	3	0	1	0
3	2	1	0	3

(4/5)



❑ 예제)

- ❖ P = (0100 0111) 이라면
- ❖ S0 : 1,4 요소 ➔ 행, 2,3 요소 ➔ 열
  - ✓ 행 00(0), 열 10(2)
- ❖ S1 : 5,8 요소 ➔ 행, 6,7 요소 ➔ 열
  - ✓ 행 01(1), 열 11 (3)

S0 =

	0	1	2	3
0	1	0	3	2
1	3	2	1	0
2	0	2	1	3
3	3	1	3	2

S1 =

	0	1	2	3
0	0	1	2	3
1	2	0	1	3
2	3	0	1	0
3	2	1	0	3

- ❖ S0 = 3(11), S1 = 3(11) 이 된다. (치환 효과)

(5/5)

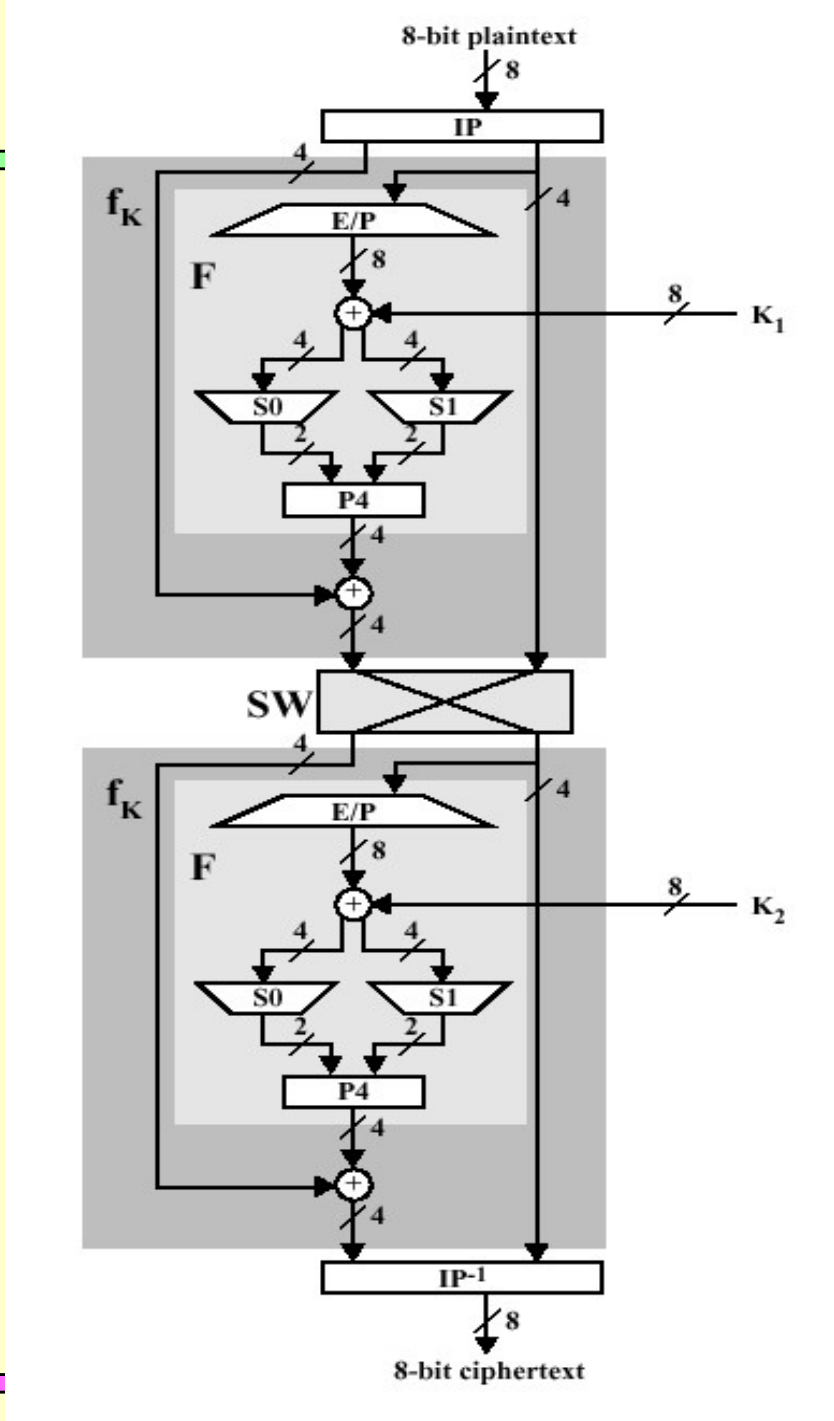
- P4 순열

P4
2 4 3 1

- P4출력은 함수 F의 출력이 된다.
- F의 출력(4비트) ⊕ L(4비트)

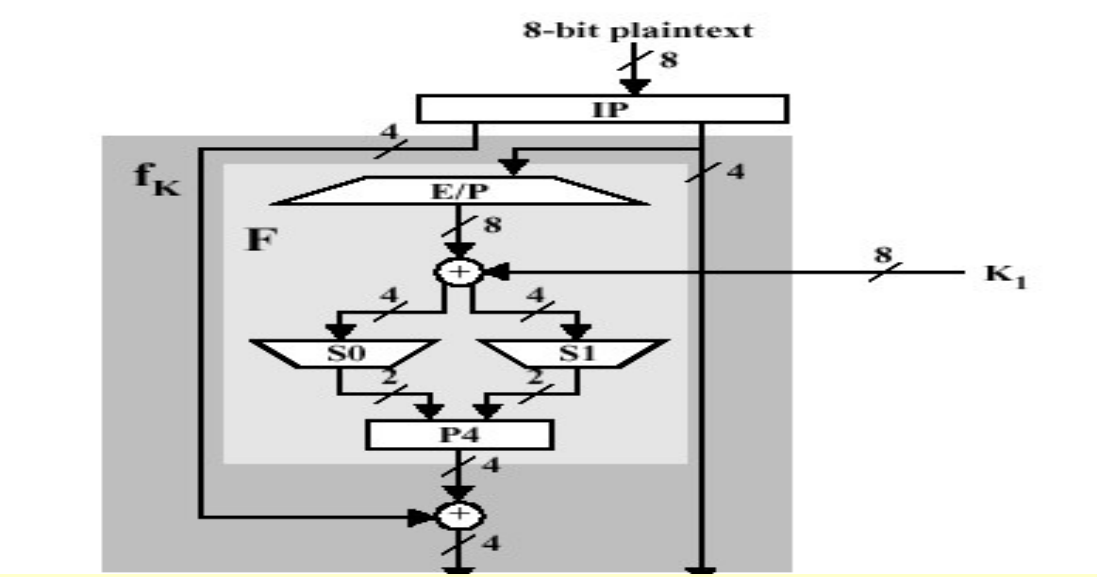
- 스위치 함수(SW)
  - ❖ f_k는 왼쪽 4비트만 변경
  - ❖ SW이용 왼쪽, 오른쪽 교환

- 두 번째 f_k에서는 K_2만 다름



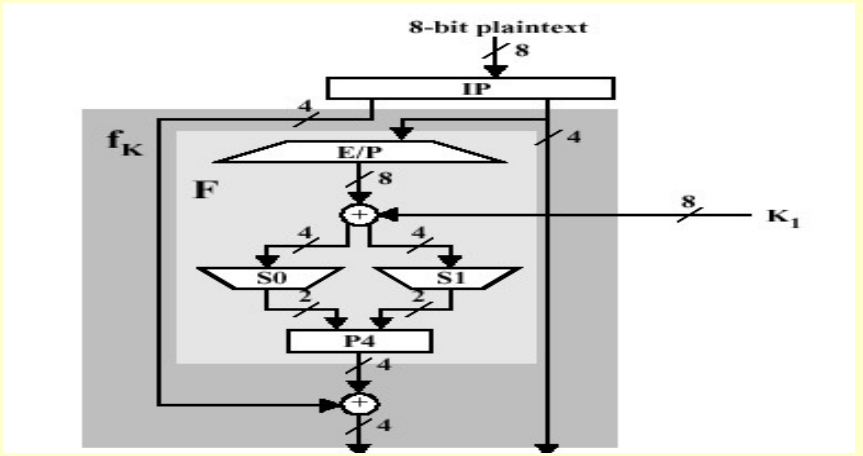
Quiz

- S-DES에서의
  - SK1 키의 값이 1 0 1 0 1 0 1 1 이고
  - 평문의 값이 1 0 1 0 1 1 0 0 인 경우
- 첫번째 라운딩 후의 결과 값은 얼마인가??



- S-DES에서의
  - SK1 키의 값이 1 0 1 0 1 0 1 1 이고
  - 평문의 값이 1 0 1 0 1 1 0 0 인 경우

첫번째 라운딩 후의 결과 값은 얼마인가??



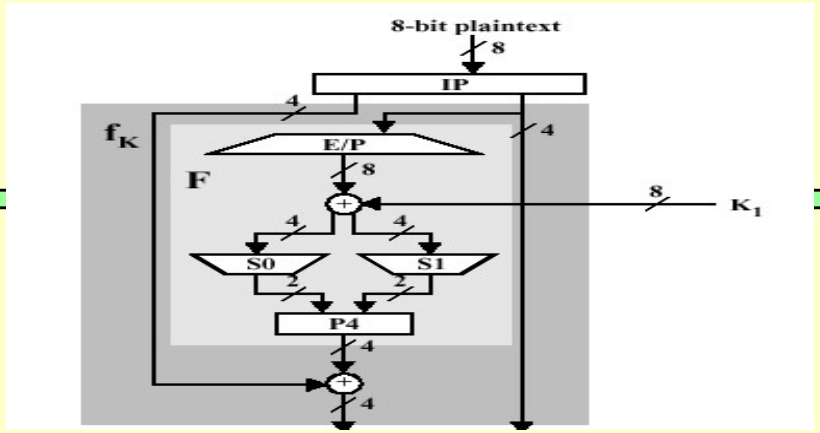
IP
2 6 3 1 4 8 5 7

- X = ( )
- IP(X) = ( )

E/P
4 1 2 3 2 3 4 1

- E/P 이후 : ( )

- 0001 0100 + 1010 1011 = ( )



❖ (,)

S0 =

	0	1	2	3
0	1	0	3	2
1	3	2	1	0
2	0	2	1	3
3	3	1	3	2

S1 =

	0	1	2	3
0	0	1	2	3
1	2	0	1	3
2	3	0	1	0
3	2	1	0	3

❖ So 박스 : 행 , 열 ➔

❖ S1 박스 : 행 , 열 ➔

❖ P4 이전 :

❖ P4 이후 :

❖ E/P 이후 :

❖ P4 출력( ) ⊕ L( ) =

P4
2 4 3 1

❖ 첫번째 라운딩 후의 결과 값 :

레포트

- S-DES(2 라운드)에서의
  - 키의 값이 **1 1 1 0 0 0 1 0 0 1** 이고
  - 평문의 값이 **0 1 1 1 0 1 1 0** 인 경우
- 2라운드 후의 암호문의 값은 얼마인가??

S-DES의 분석

- Brute-force 공격 가능
- 10비트 키 2^{10}= 1024
- 기지 평문/암호문 쌍
  - 평문: (p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8)
  - 출력 암호문: (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8)
- 미지의 키: (k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10})
- 기지 평문 공격: 각 ci는 pj와 kj의 다항식 함수gi
- 암호 알고리즘은 10개의 미지수를 갖는 8개의 비선형 방정식
- 알고리즘에서 각각의 순열과 합 연산은 선형 사상
- S박스를 통하여 비선형성을 도출
  - 선형 사상을 비선형 사상으로 변경함으로써 암호해독을 난해하게 하는 효과

레포트

- S-DES(2 라운드)에서의
  - 키의 값이 1 0 1 0 1 0 1 1 0 1 이고
  - 평문의 값이 1 1 1 1 0 0 1 0 인 경우
- 암호문의 값은 얼마인가??