2장 : Data Encryption Standard(DES)

정보보호이론

Spring 2015

2015 T.A

- 엄지은 박사 연구원
  - 프로토콜 랩 소속
    - e-mail : eom_je@korea.ac.kr

2.1 DES소개

- DES(Data Encryption Standard)
  - 1973년 미국의 연방 표준국(National Bureau of Standards, NBS, 현재의 NIST) DES 공모
  - IBM은 자사의 루시퍼(Lucifer)를 제출
  - 미 연방 표준국은 1977년 루시퍼를 수정하여 DES로 선정
    - FIPS PUB 46
    - 국가안보국(National Security Agency, NSA)는 루시퍼에서 사용된 64비트 키를 56비트로, S-Box의 형태도 변형
    - considerable controversy over its security
  - most widely used block cipher in world
  - 64-bit 데이터 블록, 56-bit 키
    - 최대 2^{56}번의 계산 필요 → 현재 2^{80} 정도의 계산이 안전
    - 현재 3-DES(triple DES)와 AES(Advanced Encryption Standard) 사용

2.2 혼돈(Confusion)과 확산(Diffusion)

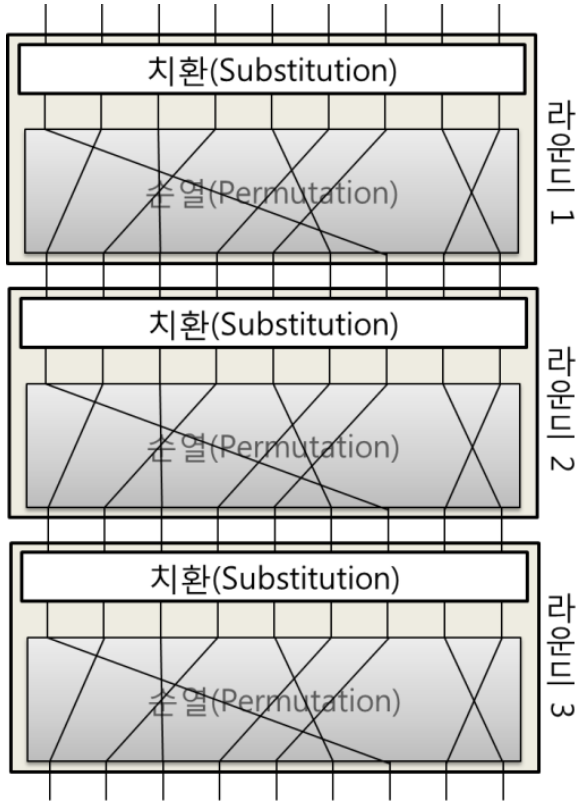
- 블록 암호 설계를 위하여 적용되는 기본적인 원리
  - 정보이론 학자인 샤논(Claude Shannon) 제안

1. **혼돈 (Confusion) :** 키와 암호문과의 관계를 감추는 성질
   - 현대 블록암호는 혼돈을 위해 치환 (Substitution) 사용
2. **확산 (Diffusion) :** 평문과 암호문과의 관계를 감추는 성질
   - 평문 한 비트의 변화가 암호문의 모든 비트에 확산
   - 주로 평문과 암호문의 통계적 성질을 감추기 위해 사용
   - DES에서는 여러 번의 순열(Permutation)을 사용하여 확산 성질을 만족하고, AES에서는 좀 더 발전된 형태인 MixColumn을 사용

Shannon

2.2 혼돈(Confusion)과 확산(Diffusion)

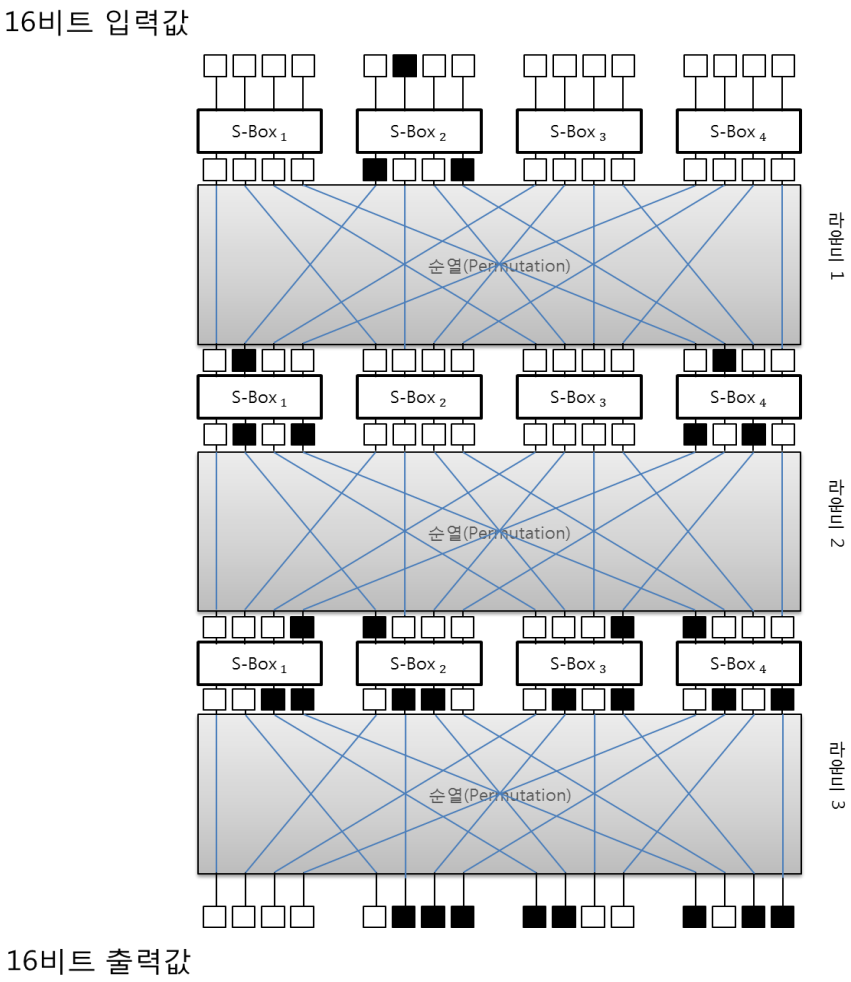
- 곱 암호(Product Cipher)
  - 샤논은 혼돈과 확산을 함께 사용하면 안전한 암호를 설계할 수 있다고 제안
    - 라운드(Round)라 불리는 부분이 여러 번 반복되는 구조
    - 한 라운드에는 치환과 순열이 사용됨



2.2 혼돈(Confusion)과 확산(Diffusion)

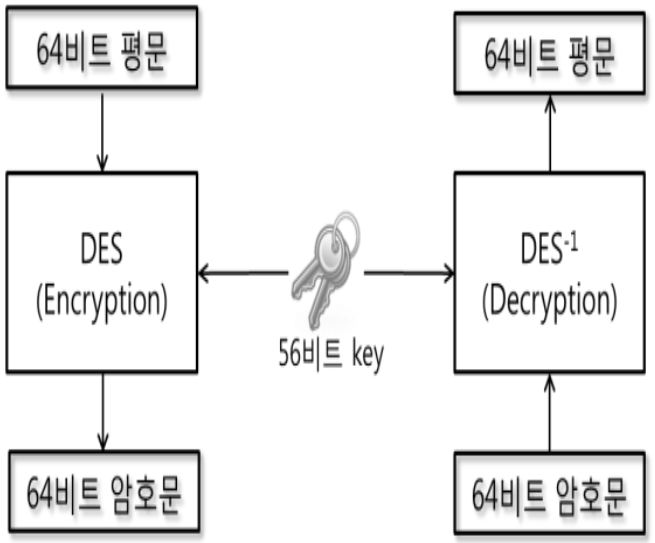
- DES : 곱 암호의 형태
  - Design principle : S-Box
    - **조건 1.** 한 비트가 상이한 두 입력 값이 S-Box에 입력되었을 때 각 입력 값에 대응되는 두 출력 값은 2비트 이상이 상이해야 한다.
    - **조건 2.** 하나의 S-Box에서 출력된 비트들은 다음 라운드에서 모든 S-Box들의 입력 값으로 확산되어 들어 갈 수 있도록 순열을 설계하여야 한다.

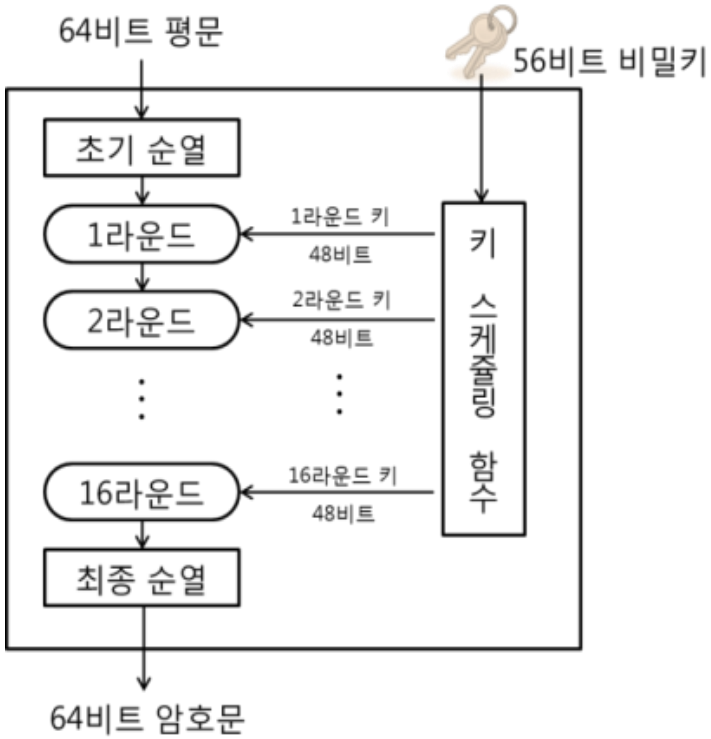
Note: 블록 사이즈가 2^n인 곱 암호가 충분한 혼돈과 확산 성질을 만족하기 위해서는 n라운드 이상으로 구성



2.3 DES 개요

- 64비트의 평문을 56비트의 키로 암호화하여 64비트의 암호문을 생성





2.4 페이스텔(Feistel) 암호구조

- 곱 암호의 두 가지 형태
  - 페이스텔(Feistel) 암호 : 가역(Invertible)요소와 비가역(Non-Invertible) 요소 모두를 사용
    - 암호화와 복호화 과정이 동일
  - 비페이스텔(Non-Feistel) 암호 : 가역 요소만 사용

Note : 전단사 함수 $f: X \rightarrow Y$에 대하여 Y에서 X로의 역관계가 존재하면 이를 역함수(Inverse Function)라고 하며 $f^{-1}: Y \rightarrow X$로 나타낸다. 가역함수는 바로 역함수가 존재하는 전단사함수를 의미

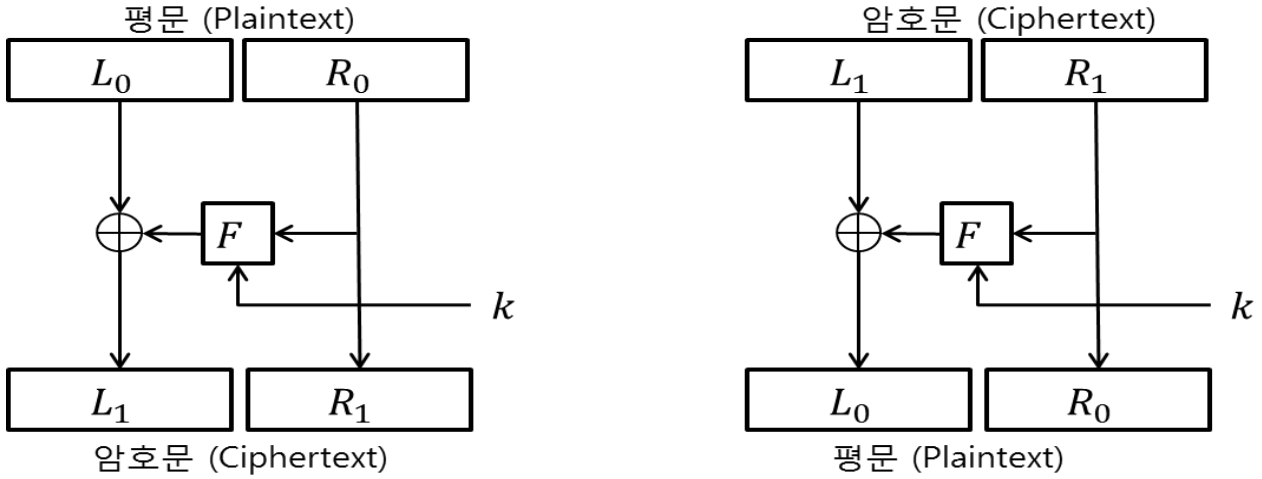
2.4 페이스텔(Feistel) 암호구조

- 1라운드 페이스텔(Feistel) 구조

암호화: $L_1 = L_0 \oplus F(k, R_0);\ R_1 = R_0$

복호화: $L_1 \oplus F(k, R_1) = L_1 \oplus F(k, R_0) = L_0 \oplus F(k, R_0) \oplus F(k, R_0)$

$= L_0;\ R_0 = R_1$



2.4 페이스텔(Feistel) 암호구조

- 2 라운드 페이스텔(Feistel) 구조

암호화: $L_1 = R_0$

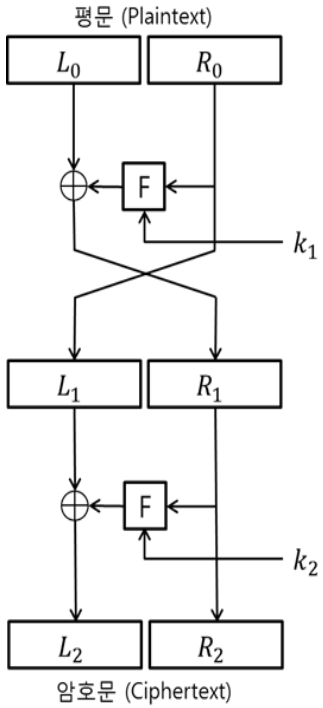
$R_1 = L_0 \oplus F(k_1, R_0)$

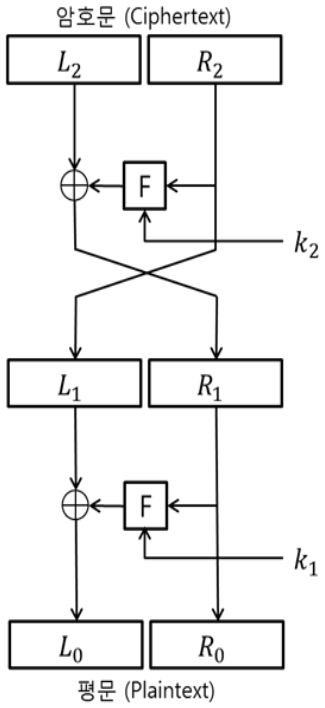
$L_2 = L_1 \oplus F(k_2, R_1)$

$R_2 = R_1$

복호화: $L_1 = R_2$

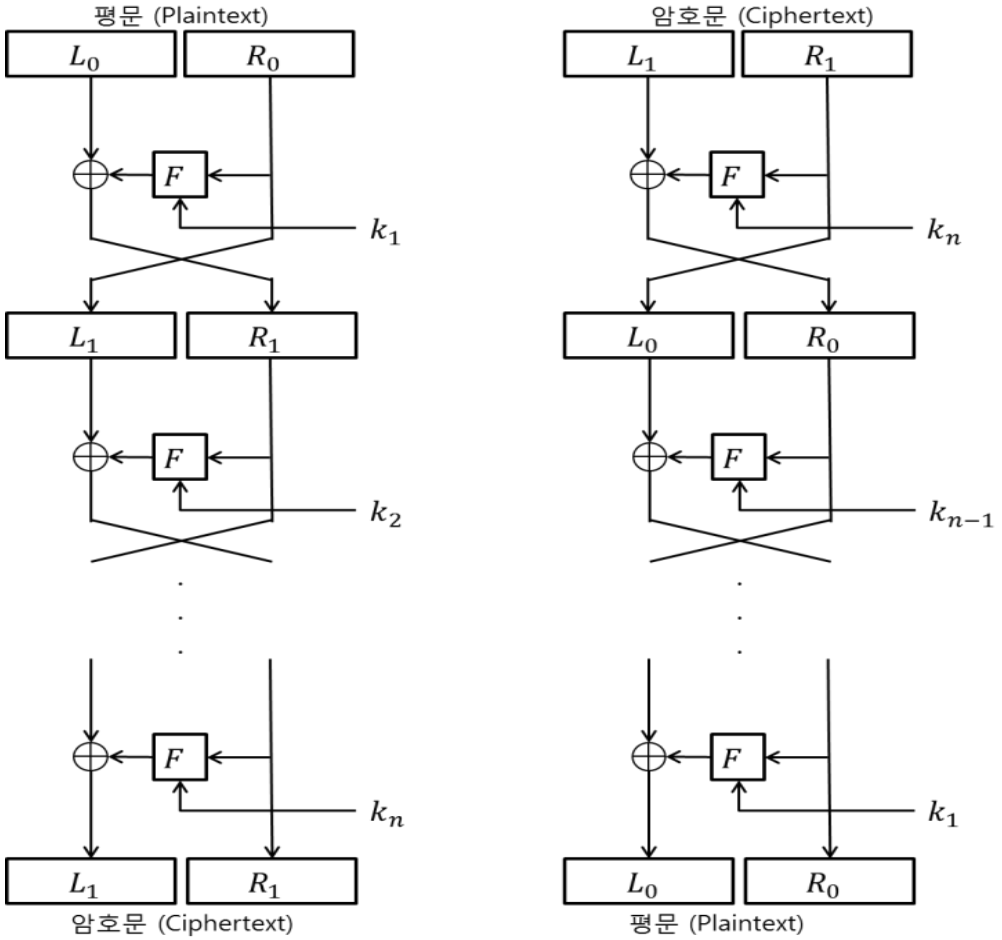
- $R_1 = L_2 \oplus F(k_2, R_2) = L_1 \oplus F(k_2, R_1) \oplus F(k_2, R_2) = L_1 \oplus F(k_2, R_2) \oplus F(k_2, R_2) = L_1$,
- $R_0 = R_1$
- $L_1 \oplus F(k_1, R_1) = R_1 \oplus F(k_1, R_0) = L_0 \oplus F(k_1, R_0) \oplus F(k_1, R_0) = L_0$





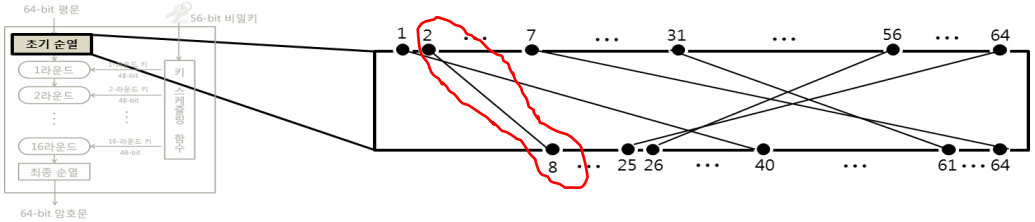
2.4 페이스텔(Feistel) 암호구조

- 다중라운드 페이스텔 (Feistel) 구조
  - 페이스텔 구조 2라운드는 SPN 구조 1라운드와 같은 안전성을 갖기 때문에, 많은 라운드 수가 필요하다.



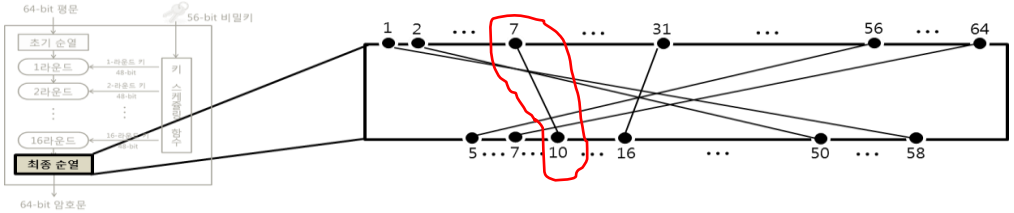
2.5 DES 상세구조

- 초기 순열 IP/ 최종 순열 FP (= IP^{-1})



초기 순열 (Initial Permutation)

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



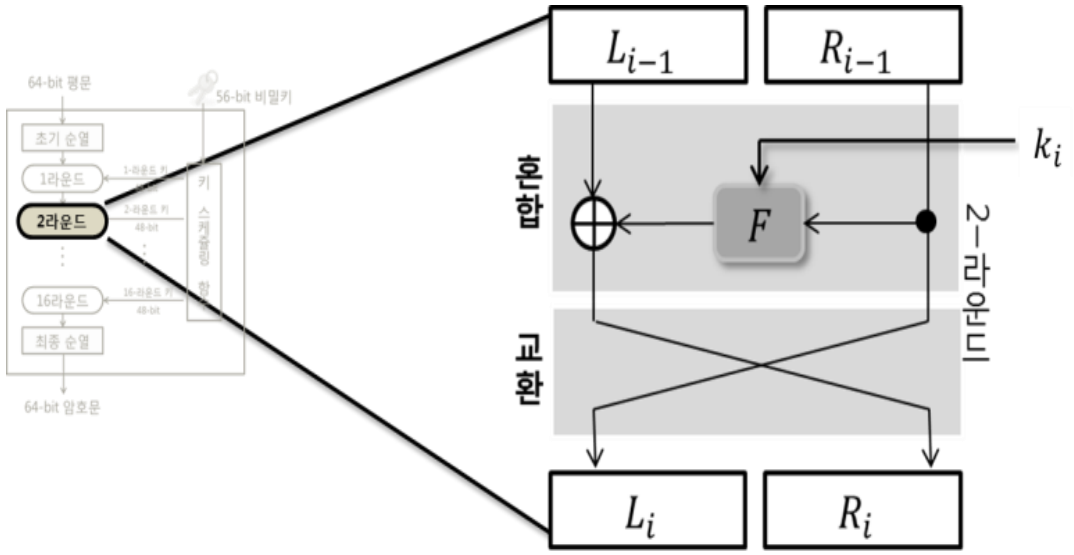
최종 순열 (Final permutation)

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

2.5 DES 상세구조

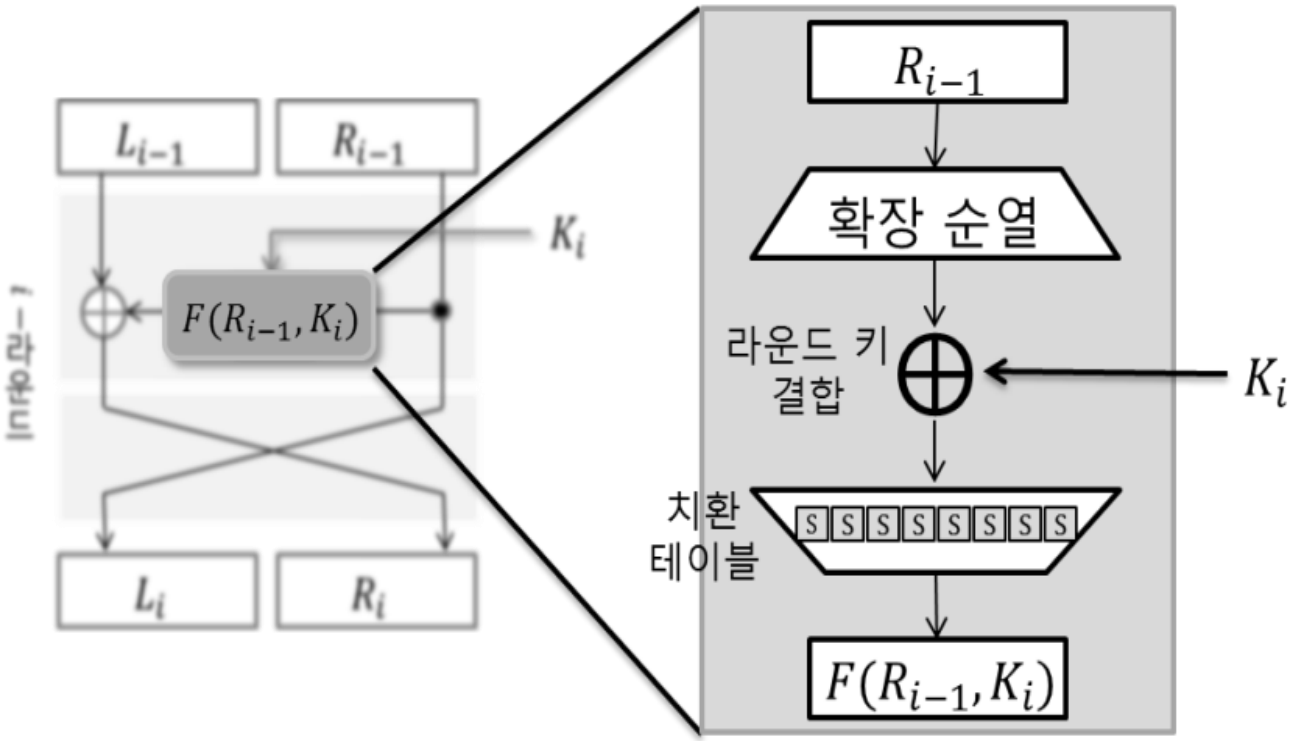
- 라운드 함수(F함수)
  - 혼합(Mixer) :오른쪽 32비트 부분과 그 라운드에 해당하는 키를 이용해 F함수 값을 계산한 후, 왼쪽 32비트 부분과 그 값을 XOR 연산 .
  - 교환(Swapper): XOR 연산을 통해 계산된 값을 오른쪽으로, 처음 입력된 값의 오른쪽 32비트를 왼쪽으로 위치를 바꿔준다.

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



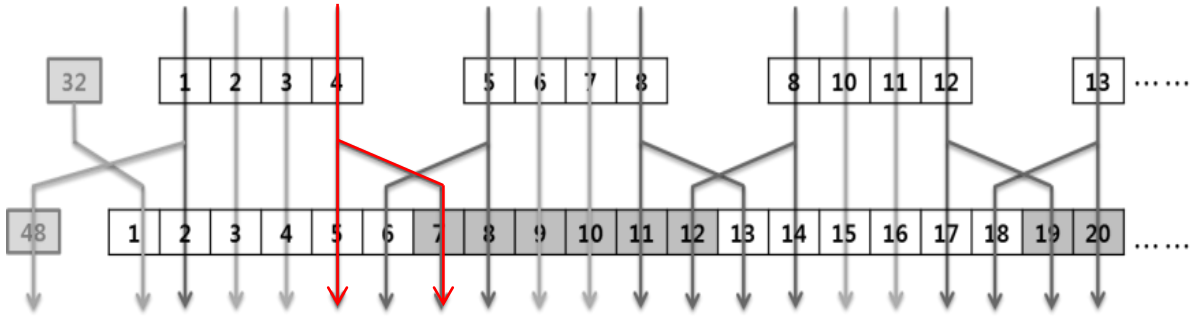
2.5 DES 상세구조

- 라운드 함수(F함수)



2.5 DES 상세구조

- 확장 순열



32비트 입력값

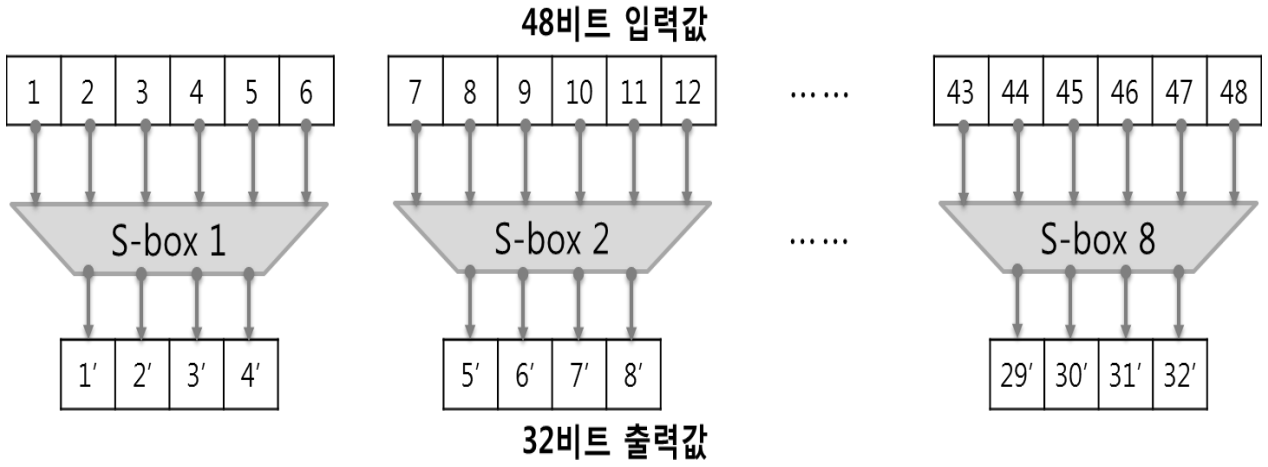
01	02	03	04
05	06	07	08
09	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32

48비트 출력값

32	01	02	03	04	05
04	05	06	07	08	09
08	09	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	01

2.5 DES 상세구조

- 라운드 키 결합(XOR) : 확장 된 48 비트 XOR 라운드 키
- 변환 테이블(S-Box)
  1. 48비트의 입력 값을 6비트씩 8개의 부분으로 나눈다.
  2. 각 6비트를 순서대로 8개의 S-Box에 입력한다.
  3. 각 S-Box는 6비트를 입력 받아 4 비트로 줄여 출력한다.

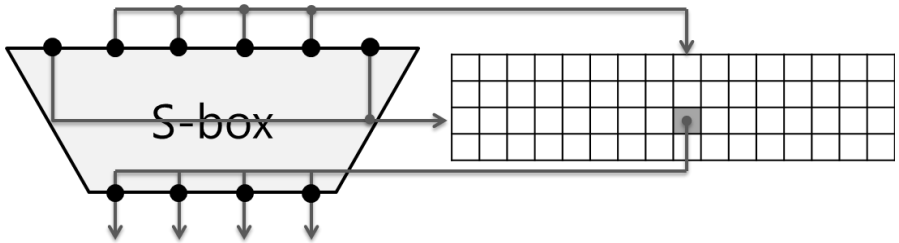


2.5 DES 상세구조

- 변환 테이블(S-Box)

  규칙 1. 입력된 6비트 중 1번째와 6번째 비트를 붙여서 십진수로 나타내면 0부터 3까지의 수 중 하나가 되는데, 이 값으로 행을 결정한다.

  규칙 2. 나머지 2번째부터 5번째까지 4 비트를 붙여서 십진수로 나타내면 0부터 15까지의 수 중 하나가 되는데, 이 값으로 열을 결정한다.



2.5 DES 상세구조

- 변환 테이블(S-Box)
  - 100111
    - Row = 11 = 3
    - Column = 0011 = 3
    - ➔ 02
  - 비선형성 :
    $S(A) \oplus S(B) \neq S(A \oplus B)$.

S-box 1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	10	03	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

S-box 2

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	01	08	14	06	11	03	04	09	07	02	13	12	00	05	10
1	03	13	04	07	15	02	08	14	12	00	01	10	06	09	11	05
2	00	14	07	11	10	04	13	01	05	08	12	06	09	03	02	15
3	13	08	10	01	03	15	04	02	11	06	07	12	00	05	14	09

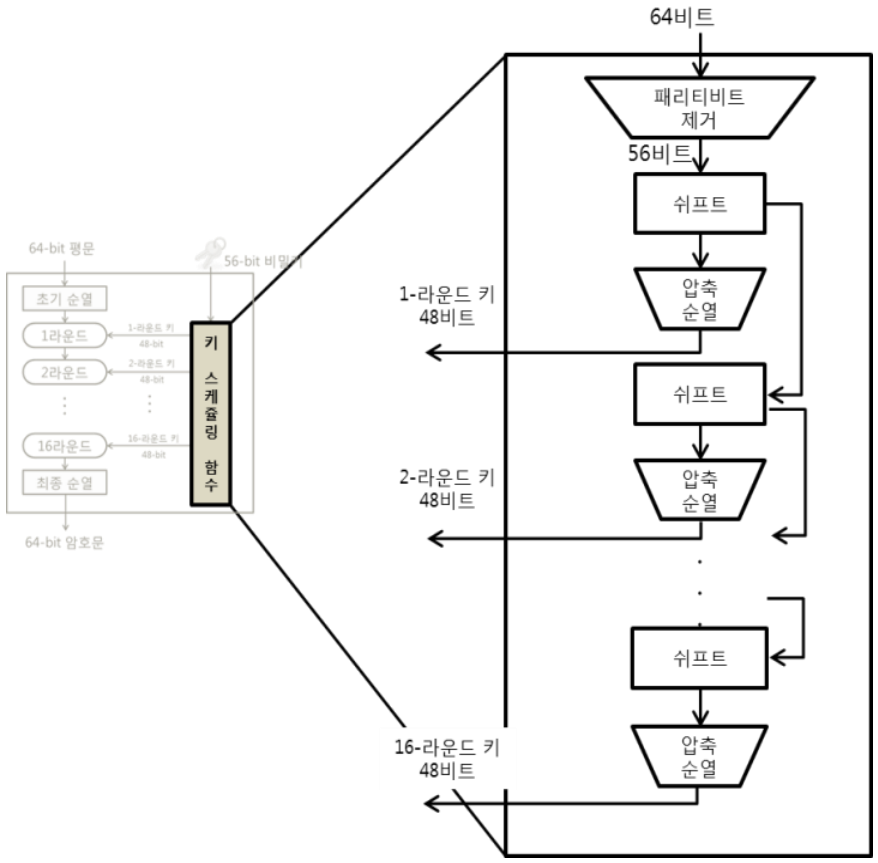
. . . .

S-box 8

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	02	08	04	06	15	11	01	10	09	03	14	05	00	12	07
1	01	15	13	08	10	03	07	04	12	05	06	11	10	14	09	02
2	07	11	04	01	09	12	14	02	00	06	10	13	15	03	05	08
3	02	01	14	07	04	10	8	13	15	12	09	00	03	05	06	11

2.5 DES 상세구조

- 키 스케줄링
  - 함수는 64비트인 비밀키를 입력으로 받아 16개의 48비트 라운드 키를 출력



2.5 DES 상세구조

- 패리티 비트 제거(Parity Bit Drop)
  - DES 암호의 64비트 비밀키 중 8개의 패리티 비트(8의 배수 비트)는 키의 안전성에 기여하지 않기 때문에, 일반적으로 DES 암호는56비트 비밀키를 사용한다

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

패리티 비트 제거 표

2.5 DES 상세구조

- 쉬프트(Shift)

$$X = x_l x_{l-1} \dots x_0,$$
$$\textbf{Cyclic Shift}(X, k) = X', \qquad i' \equiv i + k \ (mod\ l)$$
$$X' = x_{l'} x_{(l-1)'} \dots x_0$$

표 2.2 라운드 별 쉬프트 비트 표

라운드	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
횟수	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

2.5 DES 상세구조

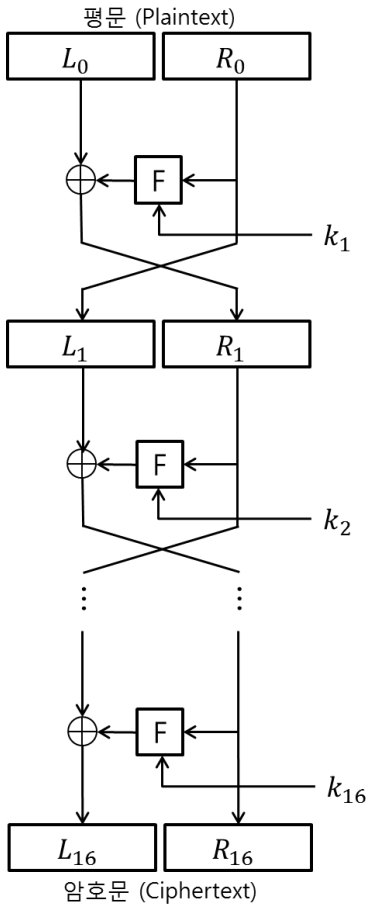
- 압축 순열(Compression P-Box)

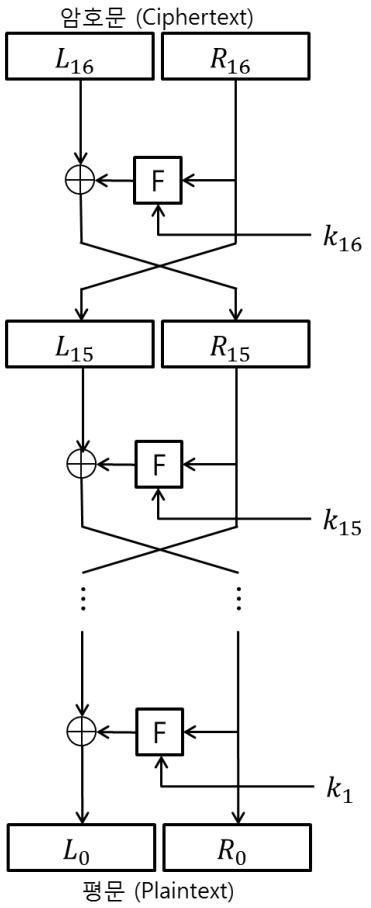
표 2.3 압축 순열 표(Compression P-Box)

14	17	11	24	01	05	03	28
15	06	21	10	23	19	12	04
26	08	16	07	27	20	13	02
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

2.6 DES의 복호화

- 암호화 과정과 동일한 알고리즘에 라운드 키를 역순으로 입력 받아 동작

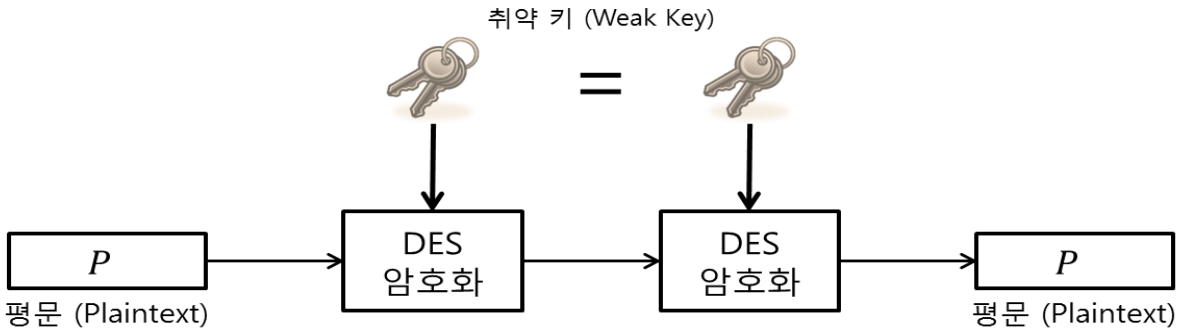




2.7 DES분석

- 취약키(Weak Key)

최초 키 스케쥴링에 입력되는 64비트	[표 2.1]의 패리티 비트 제거표를 사용한 후 56비트
0101 0101 0101 0101	0000000 0000000
1F1F 1F1F 0E0E 0E0E	0000000 FFFFFFF
E0E0 E0E0 F1F1 F1F1	FFFFFFF 0000000
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFF

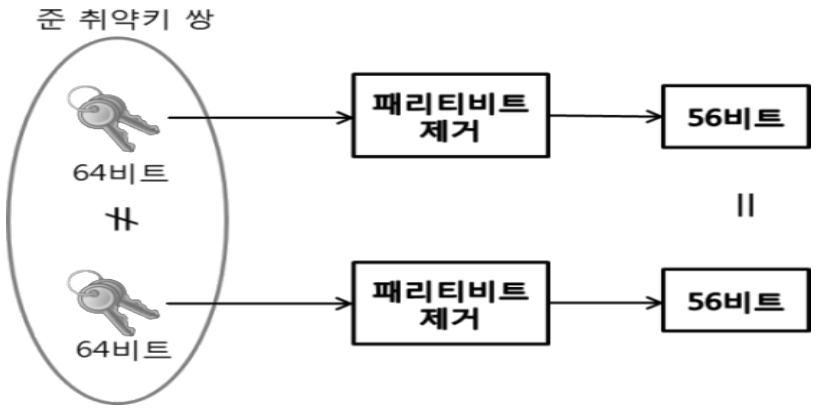


2.7 DES분석

- 준 취약키(Semi-Weak Key):
  - 2 종류의 라운드키 생성
- 가능한 준 취약키
  - 4 종류의 라운드키 생성

DES에는 취약키 4개, 준 취약키 12개, 가능한 취약키 48개가 존재하지만 사용 가능한 2^{56}개의 키 중에 이러한 키들을 선택할 확률은 $64(= 4 + 12 + 48)/2^{56} = 8.8 \text{ x } 10^{-16}$

01FE 01FE 01FE 01FE	FE01 FE01 FE01 FE01
1FE0 1FE0 0EF1 0EF1	E01F E01F F10E F10E
0EF0 0EF1 01F1 01F1	E001 E001 F101 F101
1FFE 1FFE 0EFE 0EFE	FE1F FE1F FE0E FE0E
011F 011F 010E 010E	1F01 1F01 0E01 0E01
E0FE E0FE F1FE F1FE	FEE0 FEE0 FEF1 FEF1



2.7 DES분석

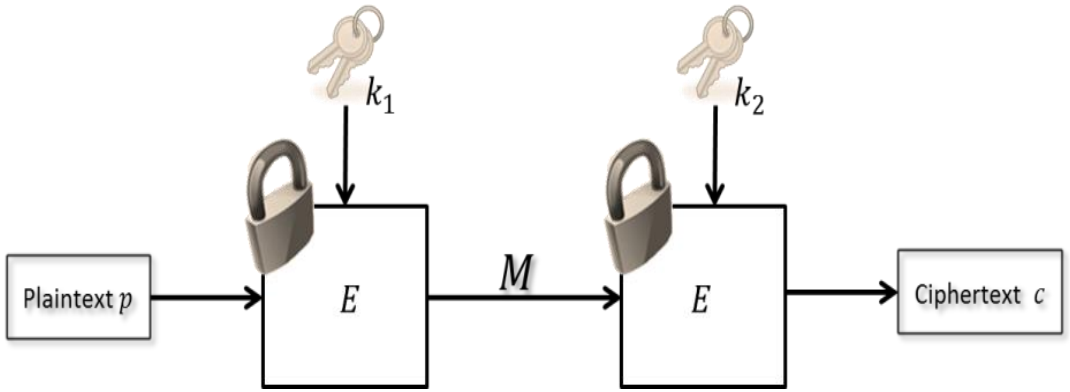
- 전사적 공격(Brute Force Attack)
  - 1981: estimated breakable in 2 days by $50M machine
  - DES Challenge I(1997): broken in 96 days by 70 000 machines, testing 7 billion keys/s (DESCHALL project)
  - DES Challenge II-1(1998): broken by distributed.net in 41 days
  - DES Challenge II-2(1998): less than 56 hours by special hardware, $250K incl design and development ("Deep Crack")
  - DES Challenge III(1999): 22 h 15 min, "Deep Crack" + 100 000 machines, testing 245 billion keys/s
  - 2007: 6.4 days, $10K hardware, 120 FPGAs (COPACOBANA project)
- 보수 특성(Complementation Property)
  - Key Complement
  - $C = E_k(P)$ → $= comp(C) = E_{comp(k)}comp(P)$
  - key domain of 2^{56}. → 2^{55}.

2.7 DES분석

- Differential Cryptanalysis
  - Biham and Shamir have shown how a 13-round iterated characteristic can break the full 16-round DES , requires an effort on the order of 2^{47} encryptions, requiring 2^{47} chosen plaintexts to be encrypted, with a considerable amount of analysis – in practise exhaustive search is still easier, even though up to 2^{55} encryptions are required for this.
- Linear Cryptanalysis
  - can attack DES with 2^{43} known plaintexts, easier but still in practise infeasible

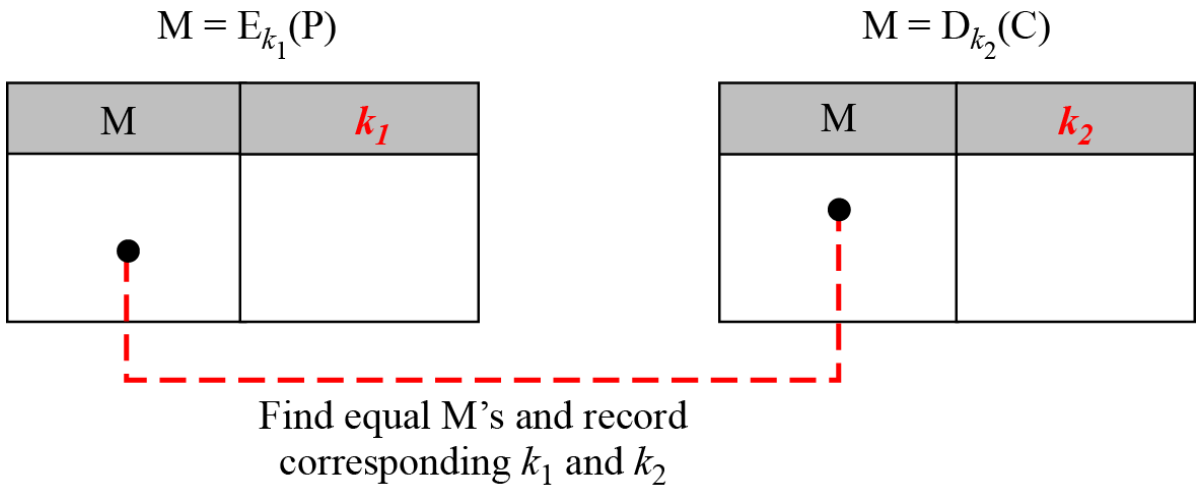
2.8 다중 DES

- 56비트인 DES암호의 짧은 키 길이를 보완하기 위해 DES를 여러 번 사용하는 다중 DES가 제안
- 2중 DES (Double DES)



2.8 다중 DES

- 2중 DES (Double DES)
  - 중간 일치 공격(Meet-in-the-Middle Attack)
    - $E_{k_1}(p) = m = D_{k_2}(c)$



2.8 다중 DES

- 3중 DES (Triple DES)