

# 제 4 장 대칭 암호



**박 종 혁 교수**

Tel: 970-6702

Email: [jhpark1@seoultech.ac.kr](mailto:jhpark1@seoultech.ac.kr)

# Chapter 4 대칭 암호

**1절 문자 암호에서 비트열 암호로**

**2절 일회용 패드-절대 해독 불가능한 암호**

**3절 DES란?**

**4절 트리플 DES**

**5절 AES 선정 과정**

**6절 Rijndael**

# 제1절 문자 암호에서 비트열 암호로

## 1.1 부호화

## 1.2 XOR

# 1.1 부호화

- 암호화에 컴퓨터 사용이 필수
- 암호화 프로그램도 평문을 비트열로 변경하고 비트열로 된 암호문을 출력
- 부호화(encoding)
  - 문자열을 비트열로 바꾸는 것

# ASCII

- 문자열 midnight 을 다음과 같은 비트열로 부호화

m → 01101101

i → 01101001

d → 01100100

n → 01101110

i → 01101001

g → 01100111

h → 01101000

t → 01110100

## 1.2 XOR

- XOR은 '익스클루시브 오아(exclusive or)', 또는 짧게 '엑스오아'라고 읽는다.

- 우리 말로는 배타적 논리합

$0 \text{ XOR } 0 = 0$  (0과 0의 XOR은 0이 된다)

$0 \text{ XOR } 1 = 1$  (0과 1의 XOR은 1이 된다)

$1 \text{ XOR } 0 = 1$  (1과 0의 XOR은 1이 된다)

$1 \text{ XOR } 1 = 0$  (1과 1의 XOR은 0이 된다)

# 한 비트의 XOR

- XOR은  $\oplus$ 이라는 기호를 써서 표현

$a \oplus b$	설명
$0 \oplus 0 = 0$	0과 0의 XOR은 0이 된다
$0 \oplus 1 = 1$	0과 1의 XOR은 1이 된다
$1 \oplus 0 = 1$	1과 0의 XOR은 1이 된다
$1 \oplus 1 = 0$	1과 1의 XOR은 0이 된다

- 같은 숫자끼리의 XOR은 반드시 0이 된다

$$0 \oplus 0 = 0$$

$$1 \oplus 1 = 0$$

# 비트열 XOR

$$\begin{array}{r} 01001100 \dots A \\ \oplus 10101010 \dots B \\ \hline 11100110 \dots A \oplus B \end{array}$$



# 비트열 XOR

$$\begin{array}{r} 1\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ \dots\ A \oplus B \\ \oplus\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ \dots\ B \end{array}$$

---

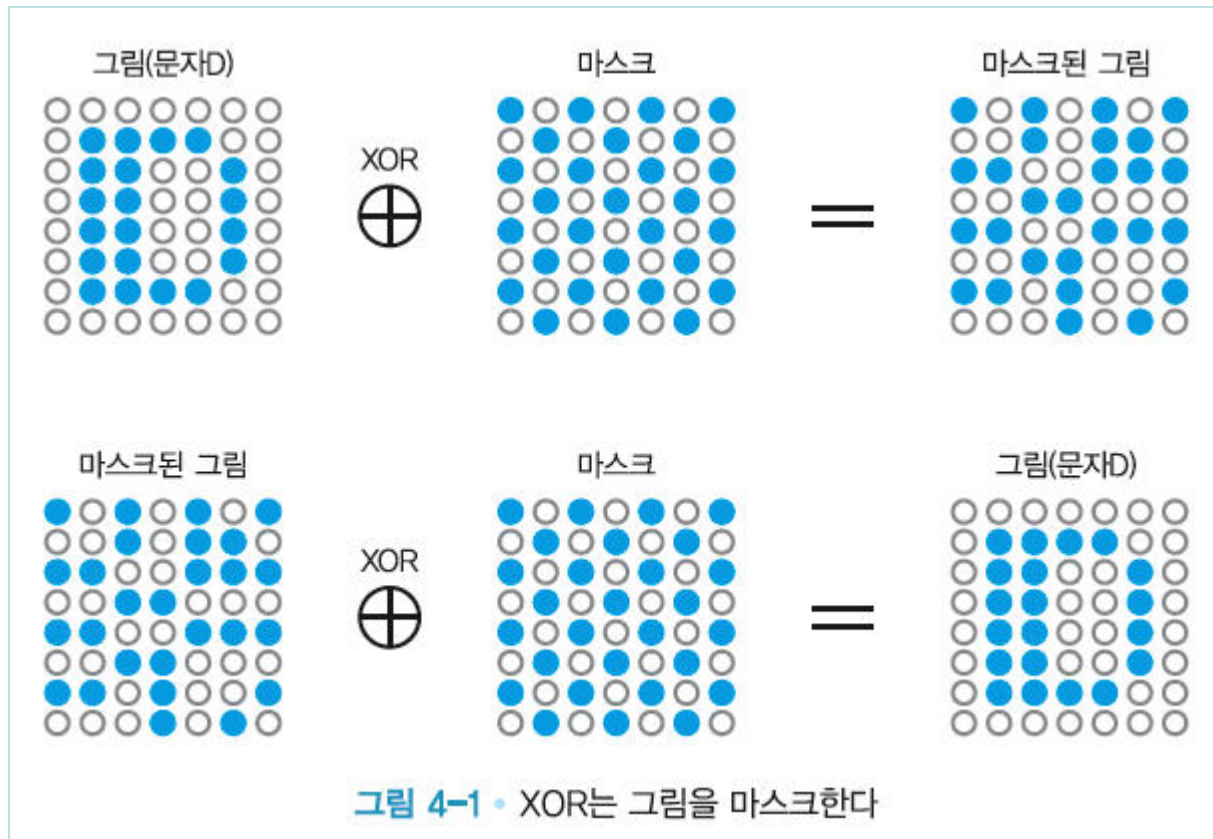
$$0\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ \dots\ A \oplus B \oplus B = A$$

(A로 돌아간다)

# 암호화/복호화의 순서와 매우 비슷

- 평문  $A$ 를 키  $B$ 로 암호화하고, 암호문  $A \oplus B$ 를 얻는다.
- 암호문  $A \oplus B$ 를, 키  $B$ 로 복호화해서 평문  $A$ 를 얻는다.

# XOR은 그림을 마스크한다



# 제2절 일회용 패드-절대 해독 불가능한 암호

## 2.1 일회용 패드란?

## 2.2 일회용 패드의 암호화

## 2.3 일회용 패드의 복호화

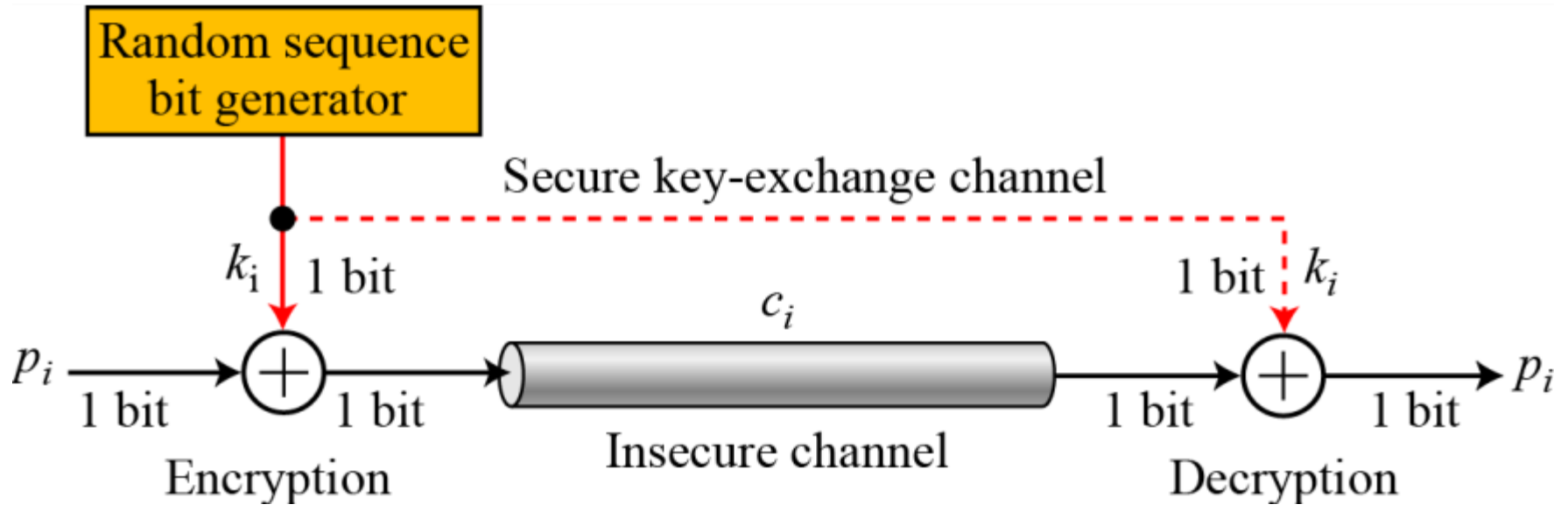
## 2.4 일회용 패드는 해독할 수 없다

## 2.1 일회용 패드란?

- 1회용 패드(one-time pad)
  - 전사공격에서 키공간을 모두 탐색하더라도 해독할 수 없는 암호

## 2.2 일회용 패드의 암호화

- 평문과 랜덤한 비트열과의 XOR만을 취하는 단순한 암호



# 일회용 패드 암호화 예

- 평문: midnight  
– ASCII로 부호화

문자	m	i	d	n	i	g	h	t
ASCII 코드	01101101	01101001	01100100	01101110	01101001	01100111	01101000	01110100
단어	m i d n i g h t							

- 키: 랜덤 비트열

키	01101011	11111010	01001000	11011000	01100101	11010101	10101111	00011100
---	----------	----------	----------	----------	----------	----------	----------	----------

# 일회용 패드 암호화 예

	01101101	01101001	01100100	01101110	01101001	01100111	01101000	01110100	midnight
$\oplus$	01101011	11111010	01001000	11011000	01100101	11010101	10101111	00011100	키
	00000110	10010011	00101100	10110110	00001100	10110010	11000111	01101000	암호문



## 2.3 일회용 패드의 복호화

- 암호문과 키의 XOR을 계산하면 평문

	00000110	10010011	00101100	10110110	00001100	10110010	11000111	01101000	암호문
$\oplus$	01101011	11111010	01001000	11011000	01100101	11010101	10101111	00011100	키
	01101101	01101001	01100100	01101110	01101001	01100111	01101000	01110100	midnight

## 2.4 일회용 패드는 해독할 수 없다

- 현실적인 시간 내에 해독이 곤란하다는 의미는 아니다.
- 키 공간 전체를 순식간에 계산할 수 있는 무한대의 계산력을 갖는 컴퓨터로도 일회용 패드는 해독할 수 없다.
- 문자열이 복호화 되었다 하더라도, 그것이 바른 평문인지 아닌지 판정할 수 없다.

# 전사 공격

- 암호문을 복호화해보면 도중에 모든 64비트 패턴이 등장한다
- 그 중에 나타날 수 있는 문자열
  - 규칙적인 문자열  
aaaaaaaa, abcdefgh, zzzzzzzz 등
  - 의미 있는 영어 단어  
midnight, onenight, mistress 등
  - 무의미한 문자열  
%Ta\_AjvX, HY(&JY!z, \$@~\*W^^), Er#f6)(%
- 따라서 어느 것이 바른 평문인지 알 수 없다
  - 즉 어떤 키를 사용하면 바르게 복호화할 수 있는지 알 수 없다

- 일회용 패드에서는 키들을 적용하여 얻어진 것이 바른 평문인지 아닌지를 판정하는 것이 불가능하다.
- 그러므로 일회용 패드를 해독할 수 없다.

# Question (1# Home Work)

- 다음 경우 각각에 대하여 one-time pad 암호의 암호문의 패턴은 무엇인가?
  - a. 평문이  $n$ 개의 0으로 구성되는 경우
  - b. 평문이  $n$ 개의 1으로 구성되는 경우
  - c. 평문이 0과 1로 교차되어 구성되는 경우
  - d. 평문이 랜덤 스트림 비트인 경우

# 제3절 DES란?

## 3.1 DES란?

## 3.2 DES 암호화/복호화

## 3.3 DES의 구조

## 3.4 Feistel 구조

## 3.1 DES 란?

- DES(Data Encryption Standard)는 1977년에 미국의 연방 정보 처리 표준 규격(FIPS)으로 채택된 대칭 암호
- 전사 공격으로 해독 할수 있는 수준

## 3.1 DES 란?

- DES는 64비트의 키를 적용하여 64비트의 평문을 64비트의 암호문으로 암호화 시키는 대칭형 블록 암호이다.
- DES 알고리즘에서는 사용하는 함수
  - 대체(substitution)
  - 치환(permutation)
- 대체와 치환은 1949년도에 Claude Shannon이 제시한 혼돈(confusion)과 확산(diffusion)이라는 두 가지 개념에 기반을 두고 있다.



# DES 콘테스트(DES Challenge)

- 1997년의 DES Challenge I
  - 96일
- 1998년의 DES Challenge II -1
  - 41일
- 1998년의 DES Challenge II -2
  - 56시간
- 1999년의 DES Challenge III
  - 22시간 15분

# • 단순 DES

- 교육용 알고리즘으로 손으로 예제를 풀 수 있음
- 단순 DES는 S-DES라고 표현
  - 8비트 평문 블록 (1 0 1 1 1 1 0 1)
  - 10비트 키를 입력

암호알고리즘

$$IP^{-1} \circ f_{K_2} \circ SW \circ f_{K_1} \circ IP$$

다른 표현

$$\text{ciphertext} = IP^{-1}(f_{K_2}(SW(f_{K_1}(IP(\text{plaintext}))))))$$

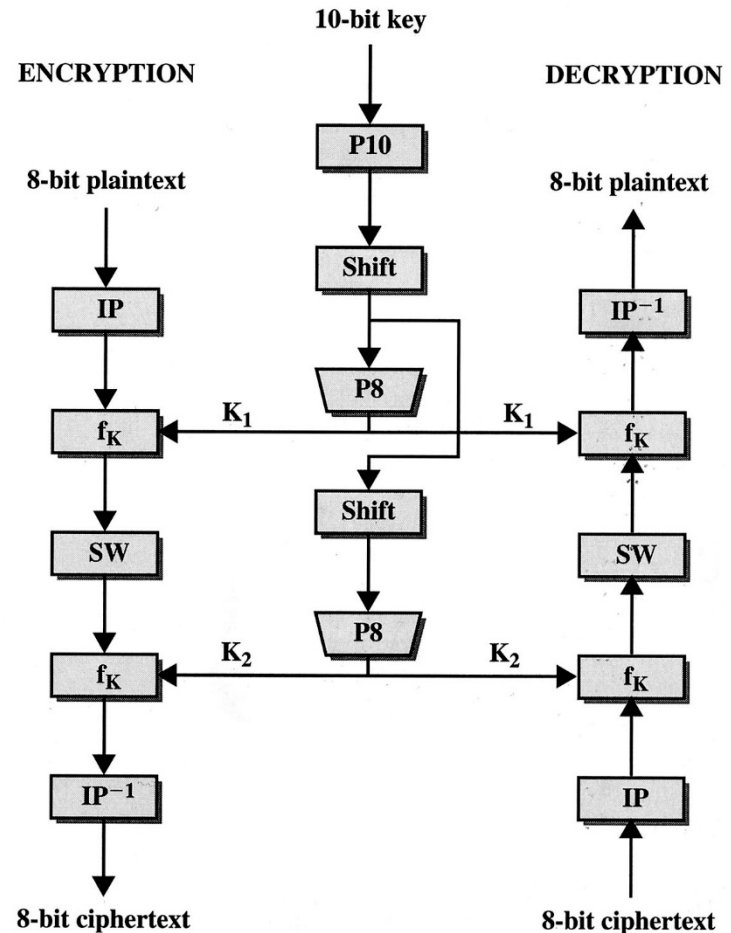
K에 대한 표현

$$K_1 = P8(\text{Shift}(P10(\text{Key})))$$

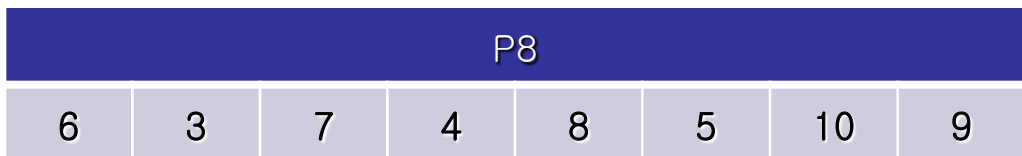
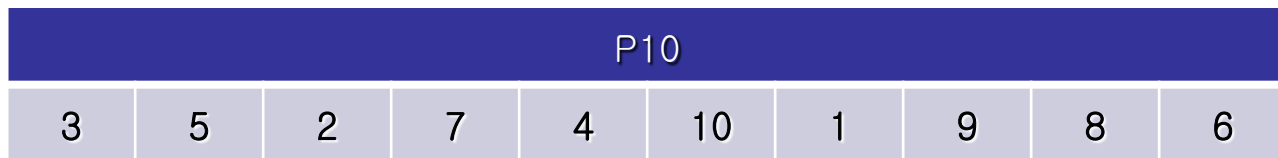
$$K_2 = P8(\text{Shift}(\text{Shift}(P10(\text{Key}))))$$

복호화 알고리즘

$$\text{plaintext} = IP^{-1}(f_{K_2}(SW(f_{K_1}(IP(\text{ciphertext}))))))$$

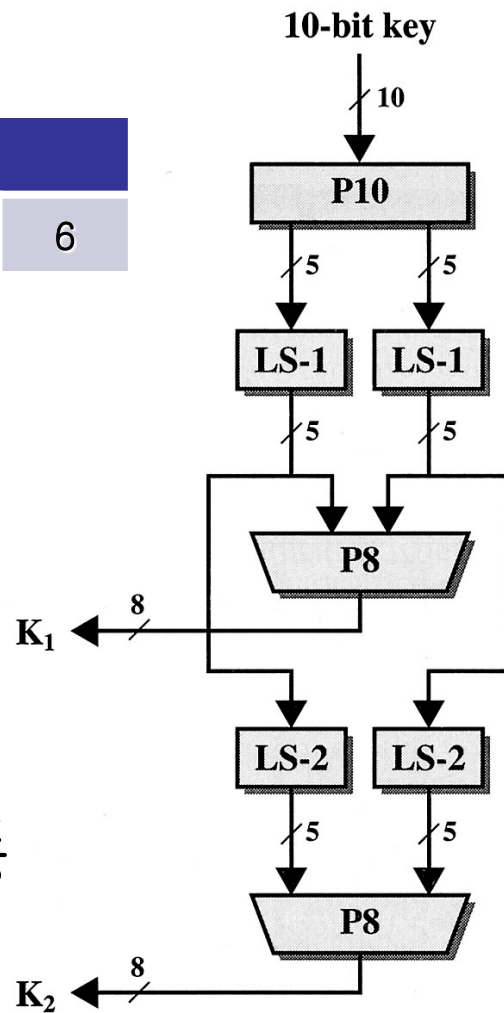


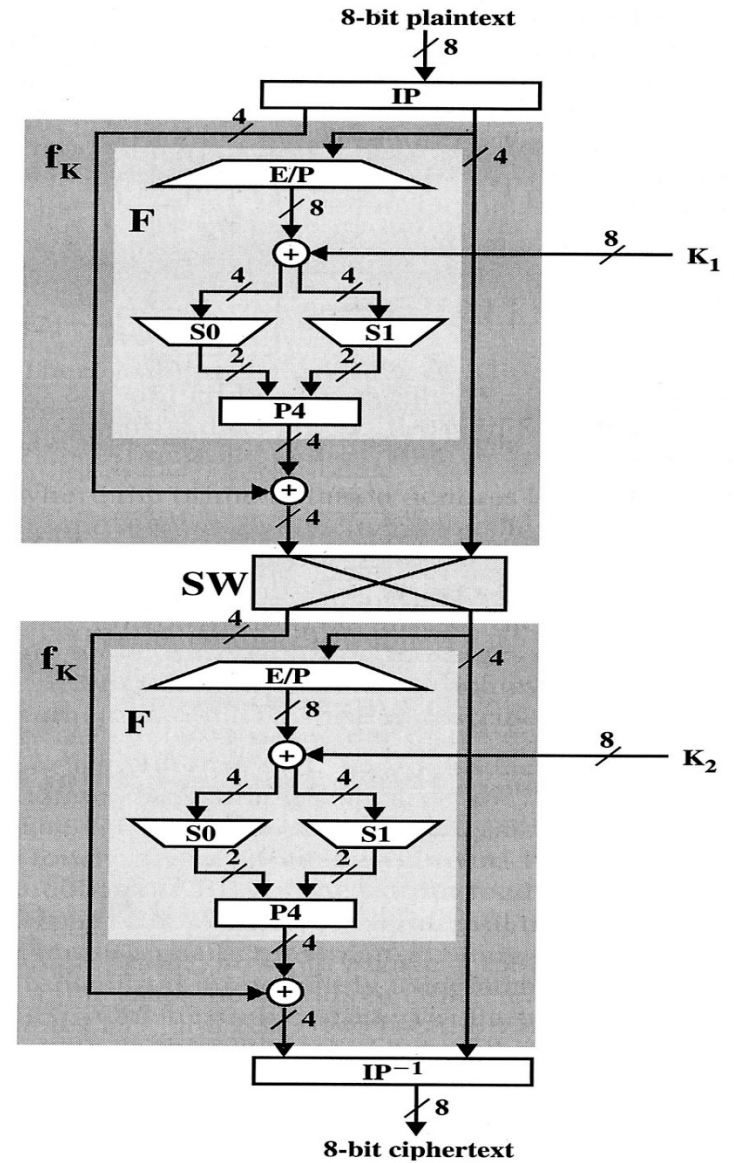
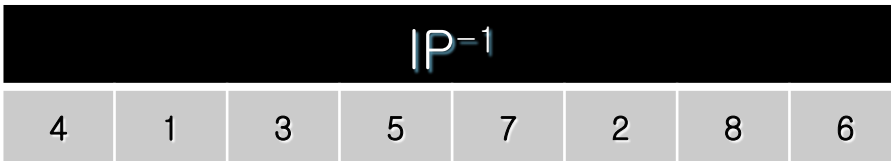
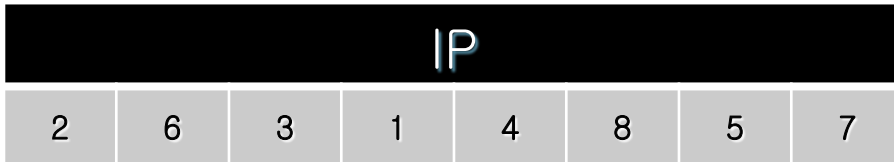
# • 키의 생성 P10



# • LS는 비트를 좌로 순환 이동

- LS-1은 한 비트 좌로 이동
- LS-2은 두 비트 좌로 이동





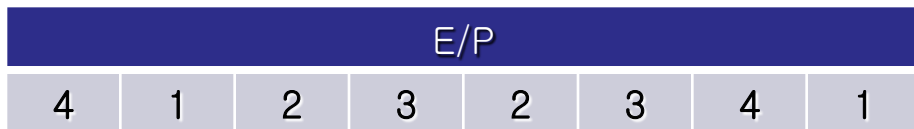
- 함수  $F$

$$f_k(L, R) = (L \oplus F(R, SK), R)$$

SK: 서브키

$\oplus$ : 비트별 배타적 논리합

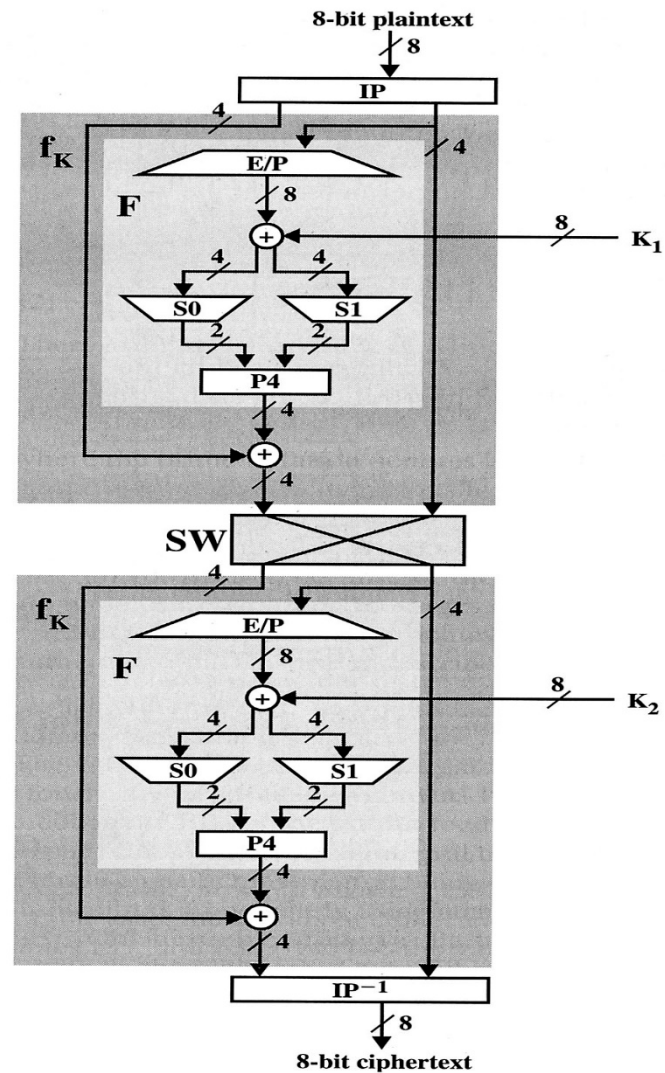
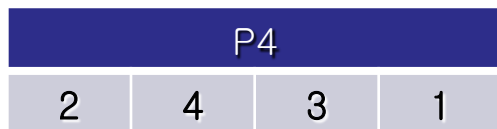
- E/P(확장/수열)



- S-box의 S0, S1

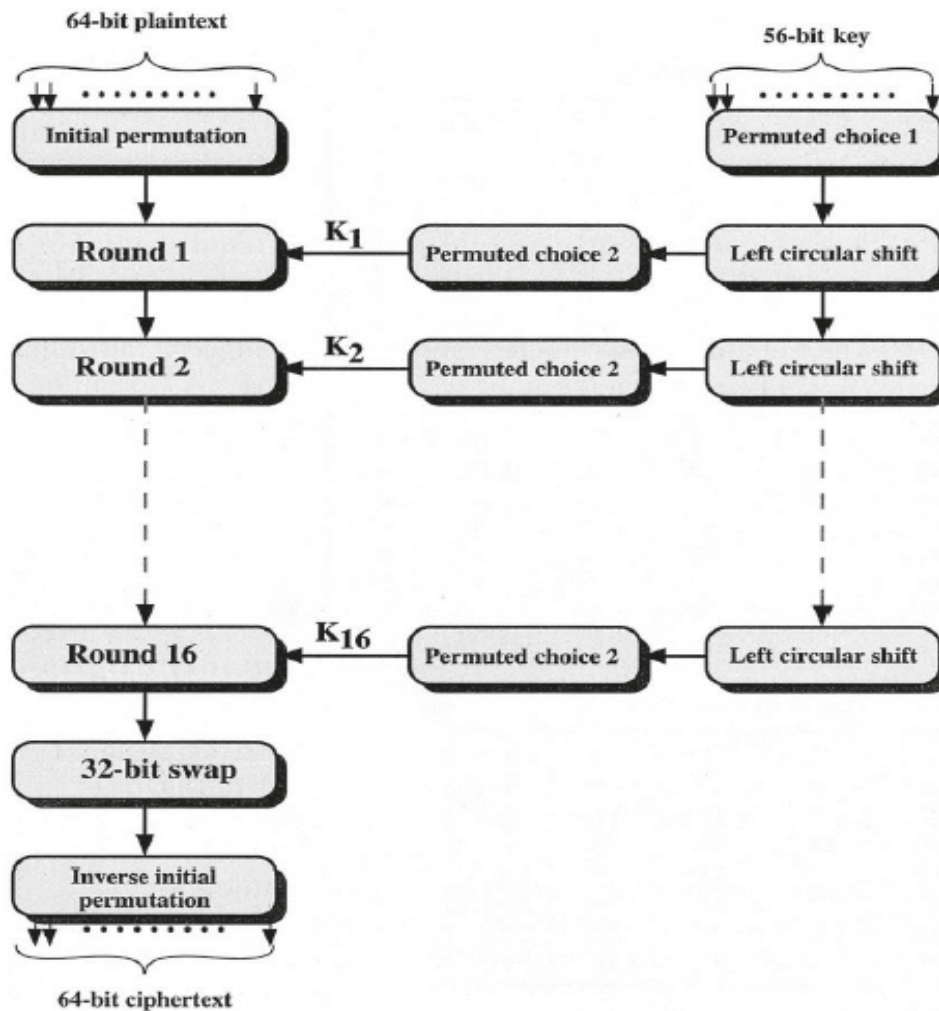
$$S_0 = \begin{matrix} & 0 & 1 & 2 & 3 \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} & \begin{pmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{pmatrix} \end{matrix} \quad S_1 = \begin{matrix} & 0 & 1 & 2 & 3 \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} & \begin{pmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{pmatrix} \end{matrix}$$

- P4



# 3.1 DES 란?

- 암호화키 56비트를 이용하여 64비트 출력으로 변환



# 3.1 DES 란?

- 초기순열(IP)

58	50	42	34	26	18	10	3
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	61	43	35	27	19	11	3
61	63	45	37	29	21	13	5
63	55	47	39	31	23	15	7

- 역 초기순열(IP<sup>-1</sup>)

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

- 확장 순열(E)

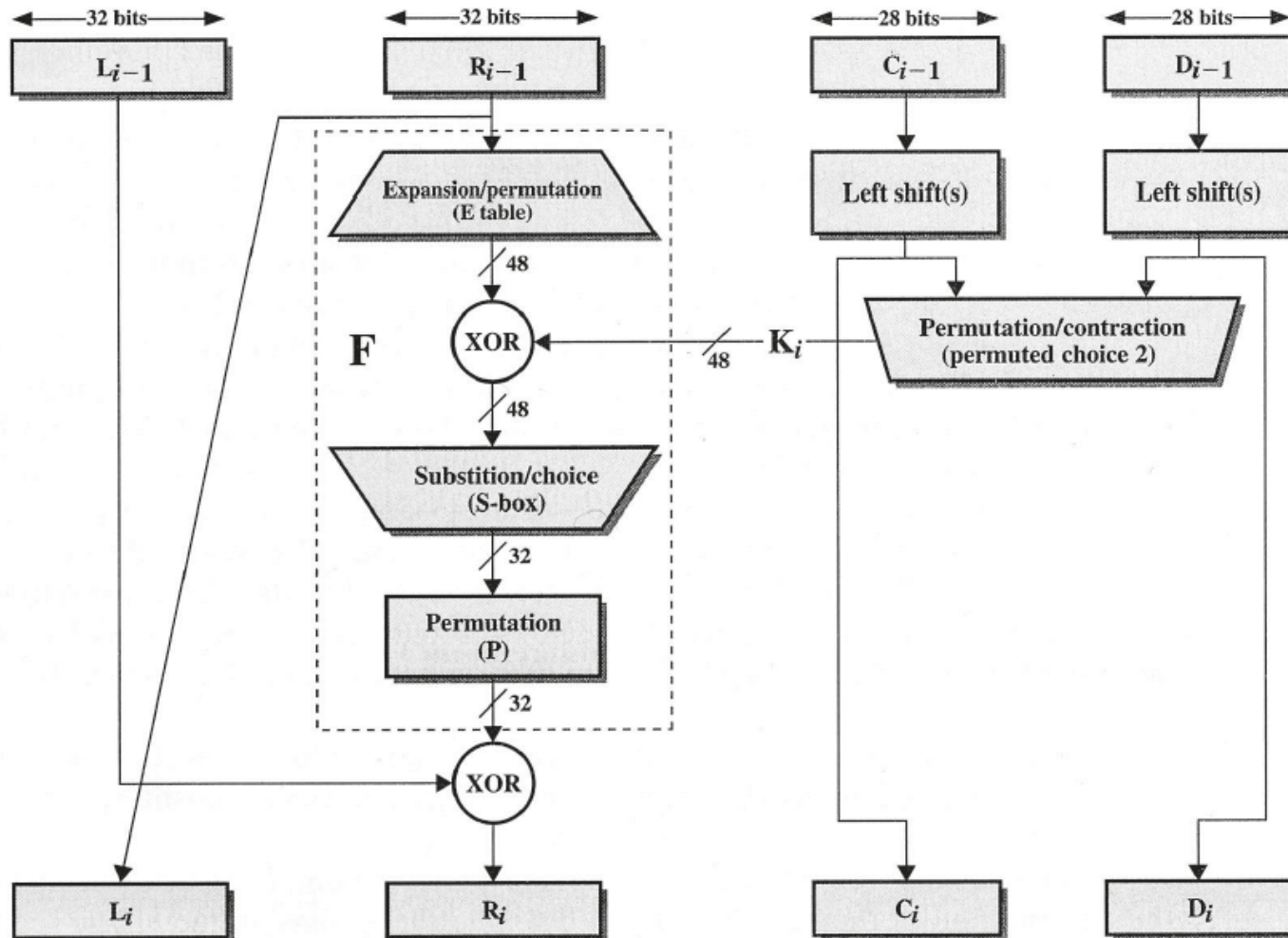
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

- 순열함수(P)

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

# 3.1 DES 란?

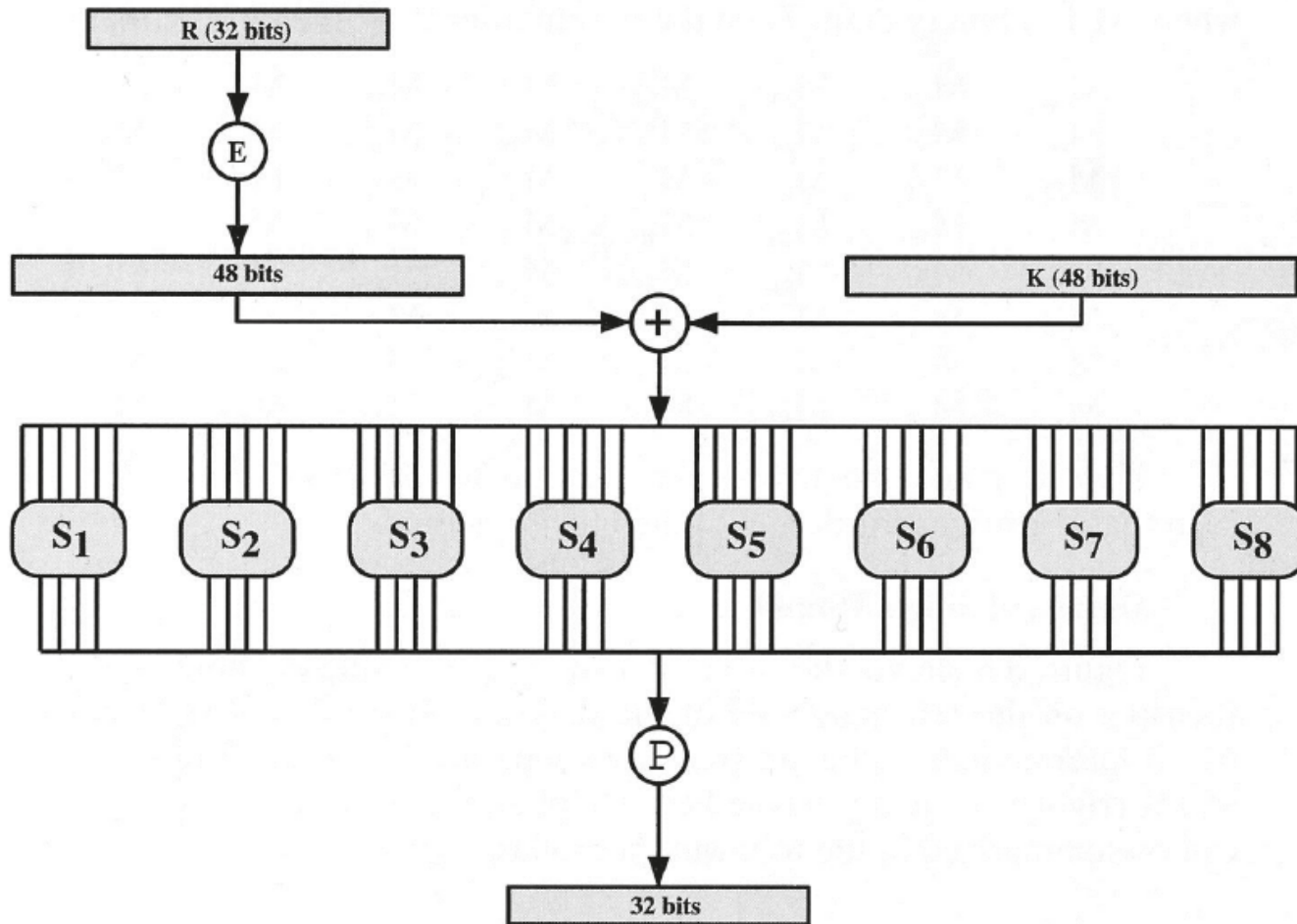
- DES 알고리즘의 단일 반복 과정





# 3.1 DES 란?

- 함수  $F(R, K)$ 의 계산



# 3.1 DES 란?

- DES의 S-박스 정의

S 1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S 2	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S 3	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S 4	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S 5	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S 6	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

# 3.1 DES 란?

- 키 생성

- DES 키의 단계별 계산표
- 순열선택1(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	63	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

- 순열선택2(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

- 쇄도우효과 (Avalanche Effect)

평문이나 키의 작은 변화가 암호문에 대하여 대하여 중요한 변화를 일으키는 암호 알고리즘의 중요한 성질

## 3.2 DES의 암호화 · 복호화

- 64비트 평문을 64비트 암호문으로 암호화하는 대칭 암호 알고리즘
- 키의 비트 길이는 56비트
- 64비트 평문(비트열)을 하나의 단위로 모아서 암호화

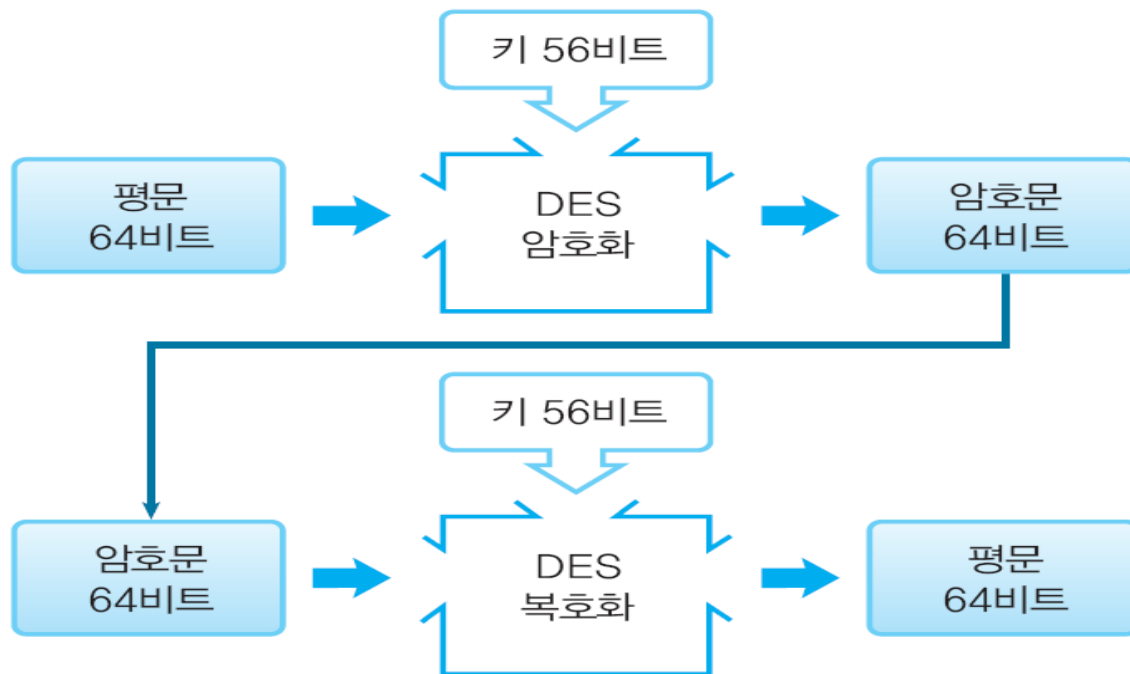


그림 4-2 DES의 암호화 · 복호화

- 블록 암호(block cipher)
  - 블록 단위로 처리를 하는 암호 알고리즘
  - 긴 비트 길이의 평문을 암호화하기 위해서는 평문을 64비트 블록으로 나누고 각각을 DES로 암호화한다

# 블록 암호 기법의 원리

- 스트림 암호

- 한 번에 1비트 혹은 1바이트의 디지털 데이터 스트림을 암호화 하는 방식

- 블록 암호 기법

- 평문 블록 전체를 가지고 같은 크기의 암호문 블록을 생성
- 모드를 이용하여 스트림 암호 기법과 동일한 효과

- Feistel 암호 방식 (**혼돈과 확산: Confusion & Diffusion**)

- Claude Shannon 소개(SHAN49): “매우 이상적인 암호는 암호문에 대한 모든 통계적 정보가 사용된 키와 독립적이어야 한다.”
- 통계적 분석에 기초한 암호 해독 방지

- **혼돈** : 키를 발견하기 어렵게 하기 위해 암호문에 대한 통계 값과 암호 키 값 사이에 관계를 가능한 복잡하게 하는 것

- **확산** : 평문의 통계적 구조가 암호문의 광범위한 통계값에 분산

- 키를 추론하기 어렵게 하기 위해 평문과 암호문 사이에 통계적인 관계를 가능한 복잡하게 만드는 것

# Feistel 암호 구조의 유도

- **n비트 블록처리: n비트 평문을 입력으로 n비트 암호문 출력**

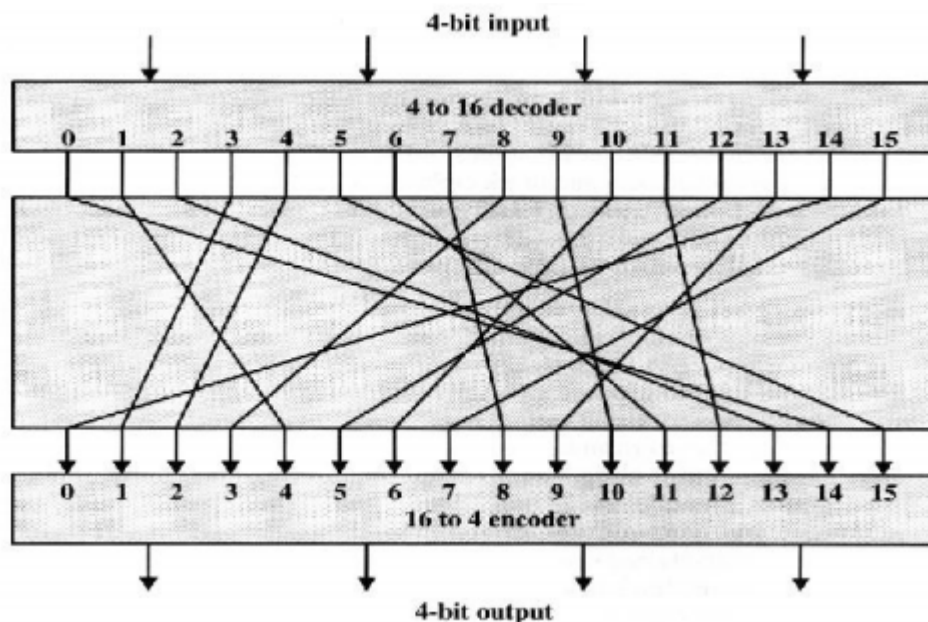
- 역으로 n 비트 암호문 입력에 대해 n비트 평문 출력

- (역의 성립: reversible, 비단수형: nonsingular)

- $2^n$  가지의 서로 다른 블록 존재 가능

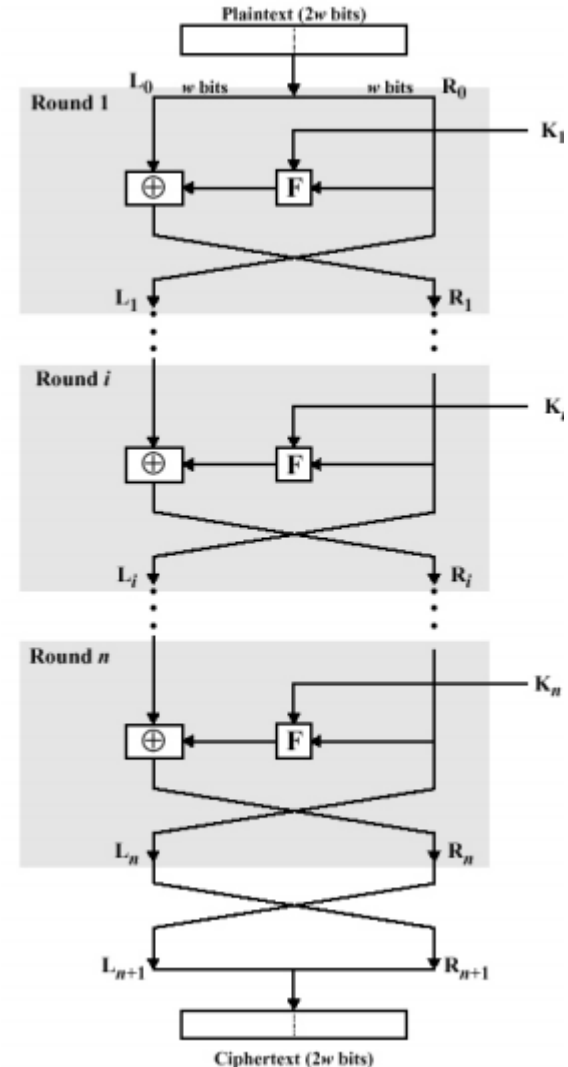
- **n 비트-n 비트 블록치환(n=4 인 경우)**

- 4비트입력으로 16개 값 중 하나 선택하고, 내부 치환에 의하여 16개 출력 값 중 하나 대응하여 4비트 출력



# Feistel 암호 구조

- 처리구조
  - 길이  $2w$  비트인 평문 블록 ( $L_0, R_0$ ) 분할 처리
  - $K$ 로 부터 유도된  $n$ 개의 키( $K_i$ ) 상용
  - $n$ 회의 동일한 반복 구조 실행
- 그림: 고전 Feistel 구조
- 하나의 반복 구조
  - 오른 쪽 반  $R_0$ 에 반복 함수  $F$  적용
    - ❖ 반복 서브키  $K_1$  적용 ( $K \neq K_i$ )
  - 왼쪽 반  $L_0$ 와 XOR(치환 작용)
  - 좌우 양쪽 결과를 교환(순열 작용)





# Feistel 네트워크의 매개 변수와 설계 특성

- 블록 크기
  - 64비트가 일반적이거나 현재는 가변적 블록 크기로 128비트로 이용함
- 키 크기
  - 64비트 또는 128비트의 크기를 이용함
- 반복 수
  - Feistel 암호 방식은 다중 반복 과정은 보안성을 증가, 일반화는 16회
- 서브키 생성 알고리즘
  - 알고리즘이 복잡할수록 암호해독이 더욱 더 어려움
- 반복 함수
  - 함수가 복잡할 수록 일반적으로 암호해독이 더욱 더 어려움
- 빠른 소프트웨어 암/복호화
  - 프로그램의 실행 속도가 관심사
- 분석의 용이성

# 제4절 트리플 DES

**4.1 트리플 DES란?**

**4.2 트리플 DES 암호화**

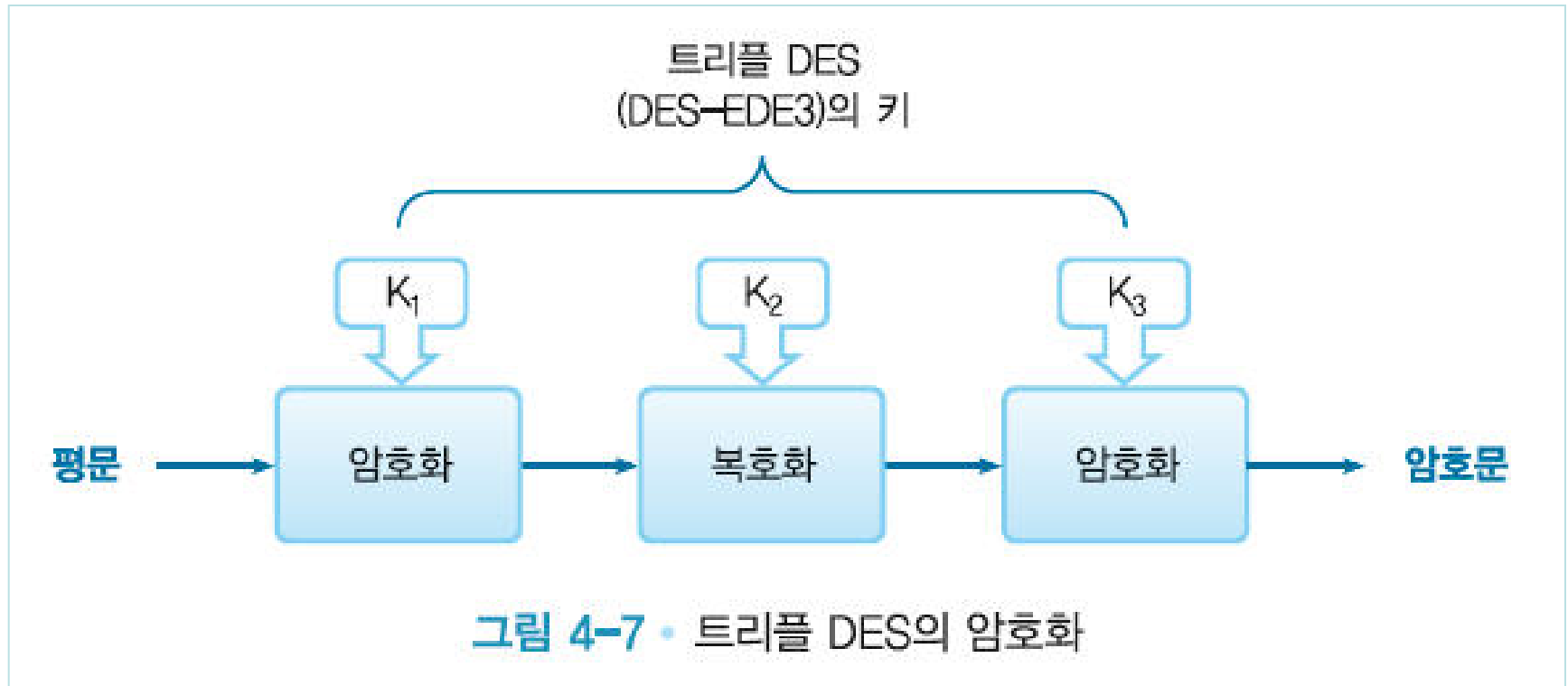
**4.3 트리플 DES 복호화**

**4.4 트리플 DES의 현황**

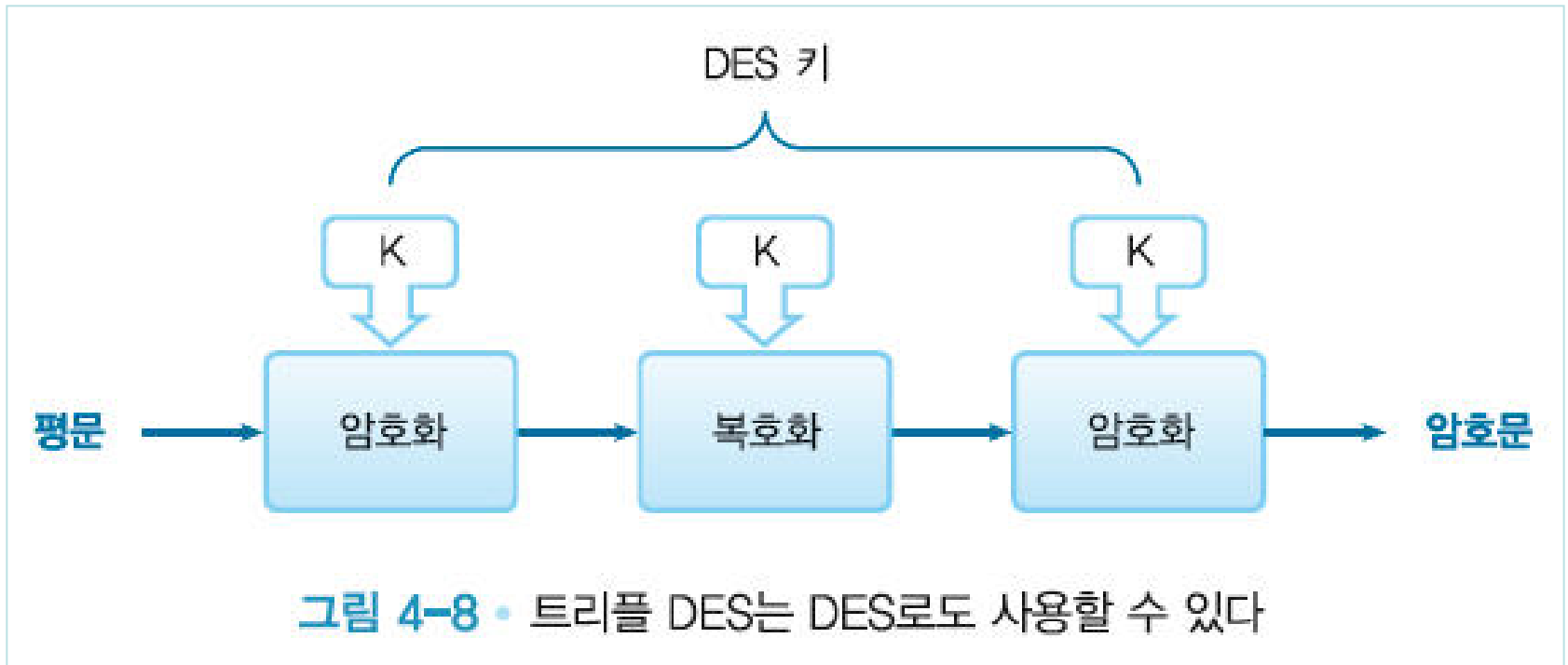
## 4.1 트리플 DES란?

- 트리플 DES(triple-DES)
- DES는 전사공격으로 현실적인 시간 내에 해독
- DES를 대신할 블록 암호가 필요
- 이를 위해 개발된 것이 트리플 DES
- DES보다 강력하도록 DES를 3단 겹치게 한 암호 알고리즘

## 4.2 트리플 DES 암호화

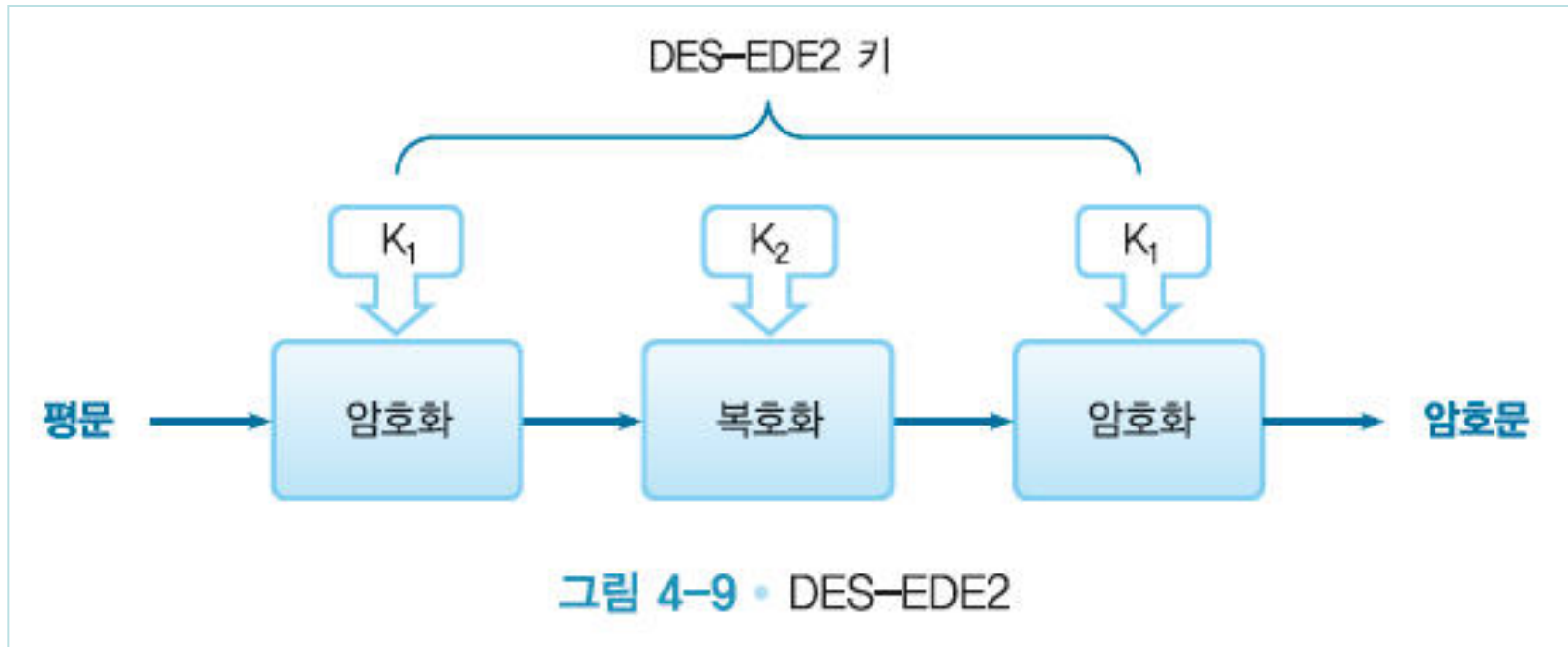


# 트리플 DES는 DES로도 사용



- DES
  - 모든 키에 같은 비트열을 사용
- DES-EDE2
  - 키1과 키3에 같은 키를 사용하고 키2에 다른 키를 사용
  - EDE는 암호화(Encryption) → 복호화(Decryption)  
→ 암호화(Encryption) 순서
- DES-EDE3
  - 키1, 키2, 키3을 모두 다른 비트열을 사용

# DES-EDE2

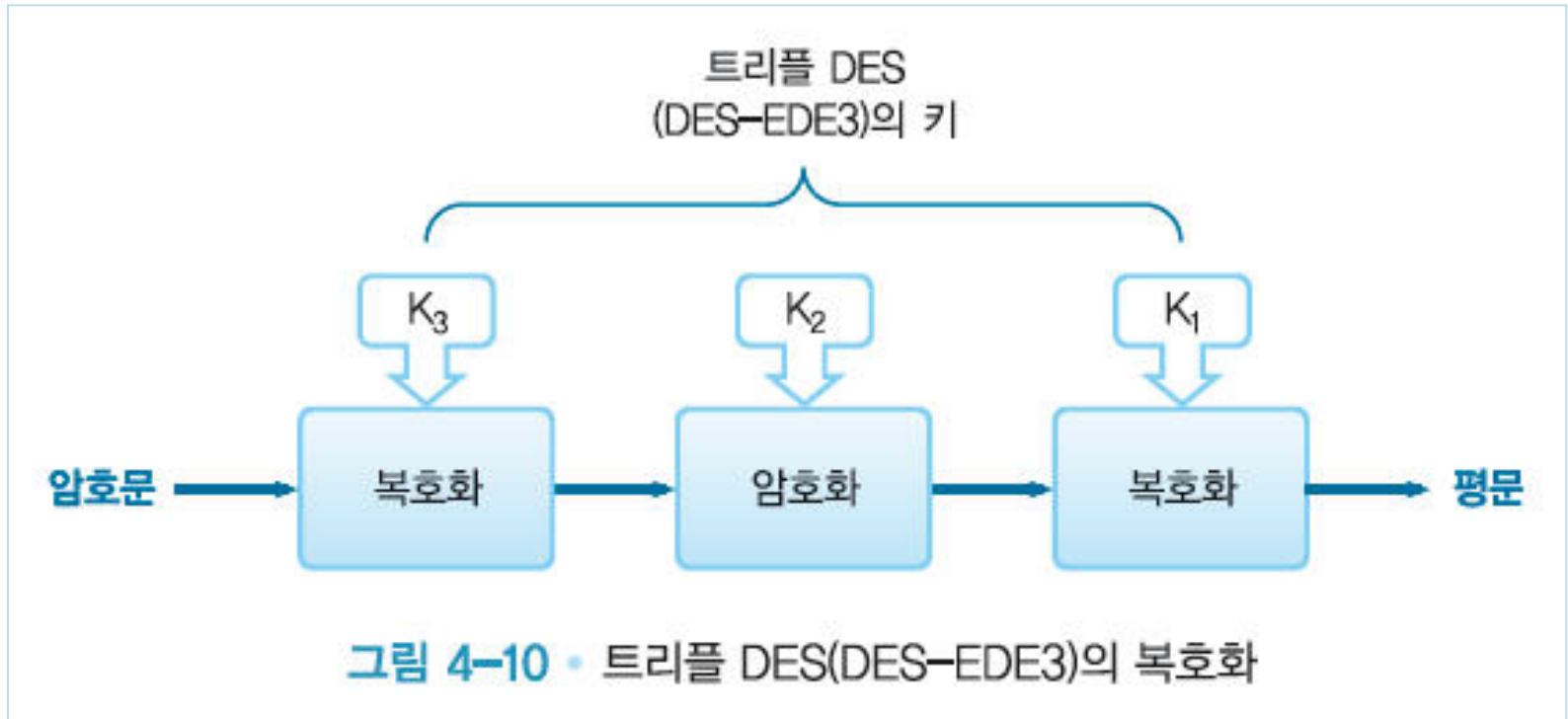


## 4.3 트리플 DES 복호화

- 암호화의 역순
- 키3, 키2, 키1의 순으로 복호화 → 암호화 → 복호화를 행한다



# 트리플 DES(DES-EDE3)의 복호화



## 4.4 트리플 DES의 현황

- 현재도 은행 등에서 아직 사용
- 처리 속도는 빠르지 않고
- 안전성 면에서도 풀려버린 사례가 있음
- 우리나라에서는 3-DES를 표준으로 정하지 않음
- 우리나라 국가표준은 SEED 및 ARIA

# 제5절 AES 선정 과정

## 5.1 AES란?

## 5.2 AES 선정 과정

## 5.3 AES 최종 후보 및 선정

## 5.1 AES란?

- AES (Advanced Encryption Standard)
  - DES를 대신한 새로운 표준 대칭 암호 알고리즘
  - AES의 후보로서 다수의 대칭 암호 알고리즘을 제안했지만, 그 중에서 **Rijndael**이라는 대칭 암호 알고리즘이 2000년에 AES로서 선정

## 5.2 AES 선정 과정

- NIST(National Institute of Standard and Technology)에서 공모
- 경쟁방식에 의한 표준화(standardization by competition)
- 조건
  - 제한 없이 무료로 이용
  - ANSI C와 Java에 의한 구현
  - 암호해독에 대한 강도의 평가
  - 암호 알고리즘 설계 규격과 프로그램 공개

## 5.3 AES 최종 후보 및 선정

- 1차 심사 통과: 15개

- CAST256, Crypton, DEAL, DFC, E2, Frog, HPC, LOKI97, Magenta, MARS, RC6, Rijndael, SAFER+, Serpent, Twofish

- 2차 심사 통과: 5개

명칭	응모자
MARS	IBM
RC6	RSA
Rijndael	Daemen, Rijmen
Serpent	Anderson, Biham, Knudsen
Twofish	Counterpane

# 제6절 Rijndael

## 6.1 Rijndael이란?

## 6.2 Rijndael의 암호화와 복호화

## 6.3 Rijndael의 해독

## 6.4 어떤 암호를 사용하면 좋은가?

## 6.1 Rijndael이란?

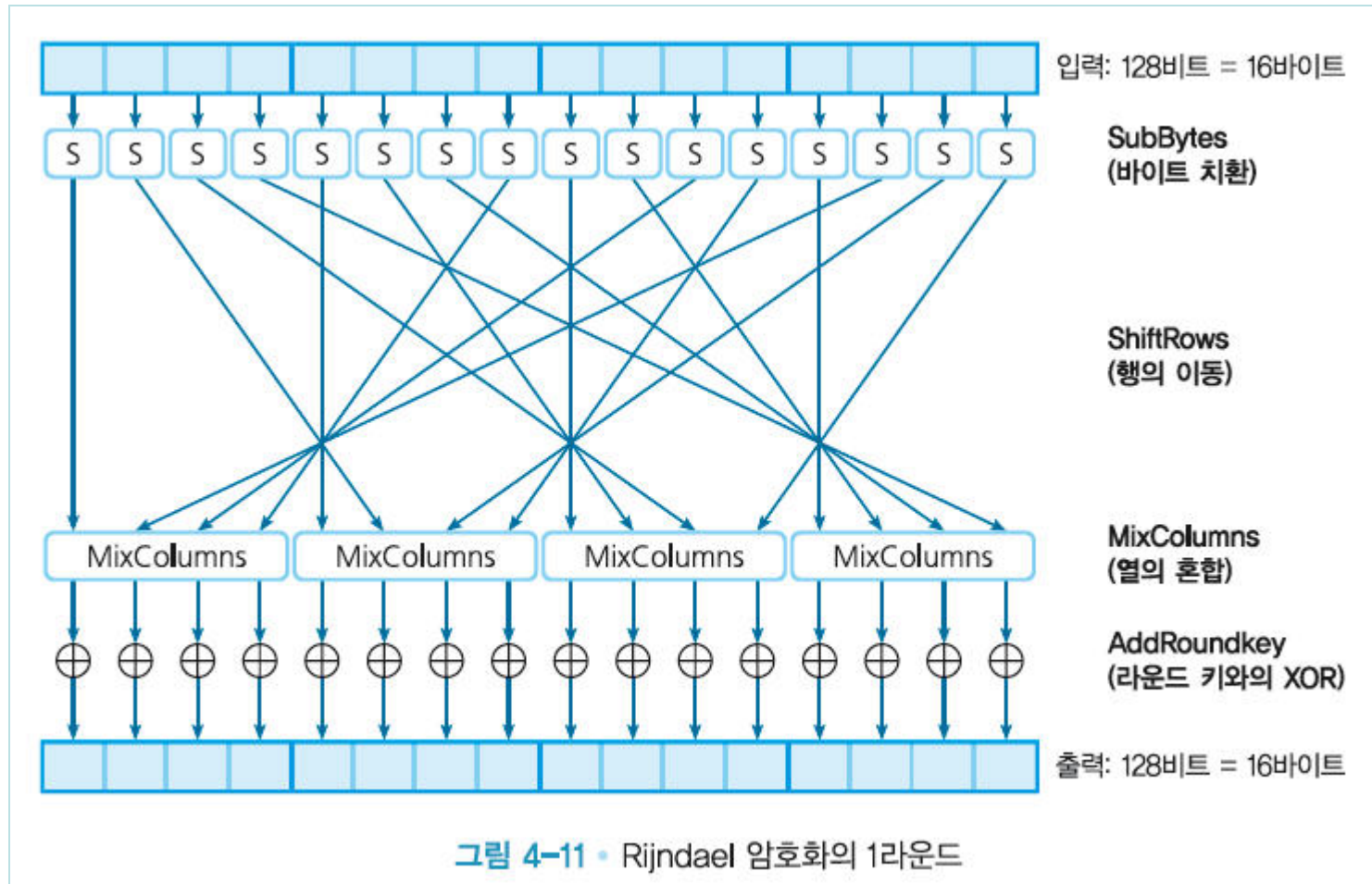
- 벨기에 연구자 Joan Daemen과 Vincent Rijmen이 설계한 블록 암호 알고리즘
- 블록 길이
  - 128비트
- 키의 비트 길이
  - 128비트
  - 192비트
  - 256비트



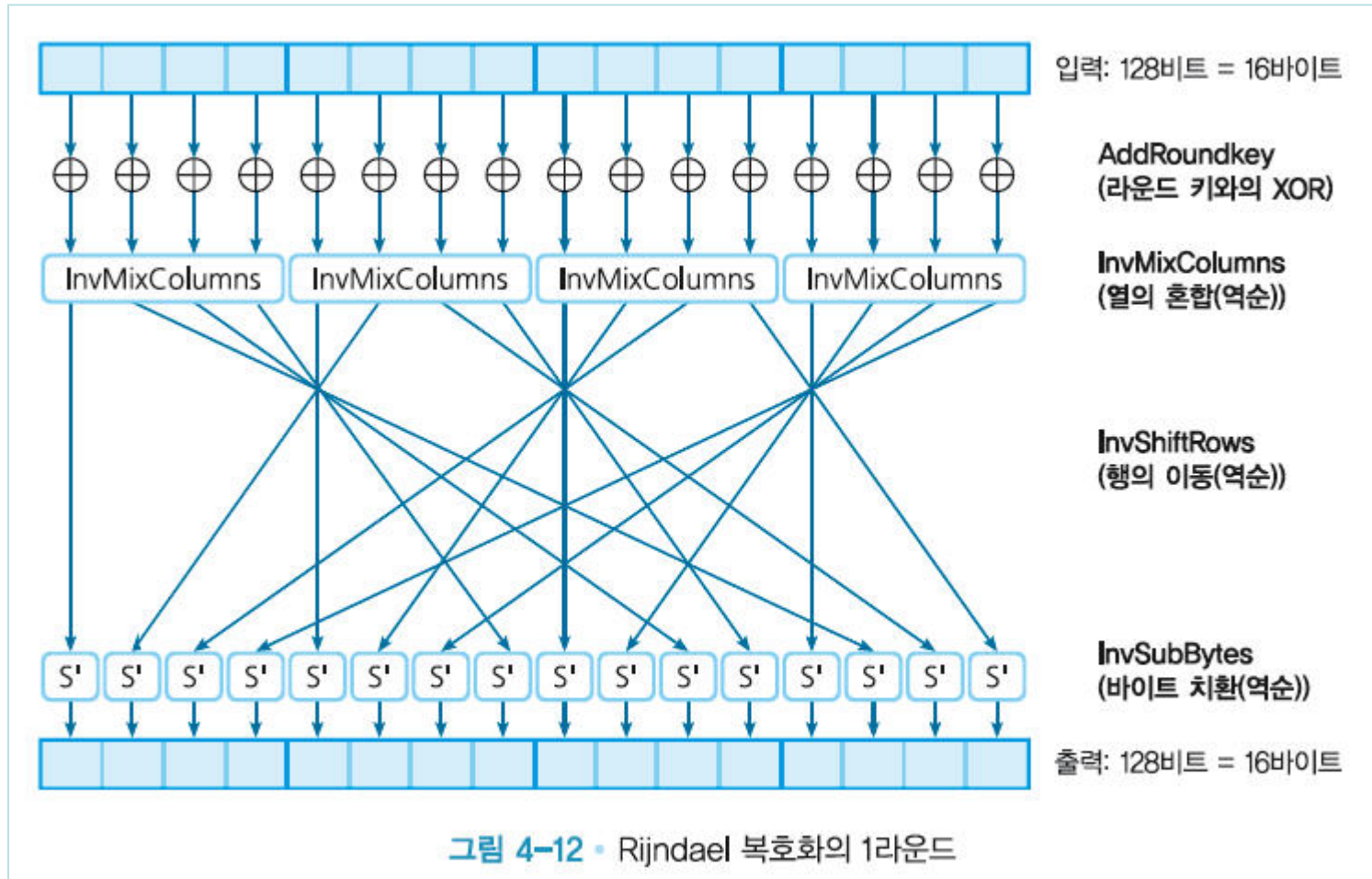
## 6.2 Rijndael의 암호화와 복호화

- 복수의 라운드(round)로 구성(10~14)
- SPN(Substitution-Permutation Network) 구조
- SubBytes
- ShiftRows
- MixColumns
- AddRoundKey

# Rijndael 암호화 1라운드



# Rijndael 복호화 1라운드



# SubByte(바이트 대체)

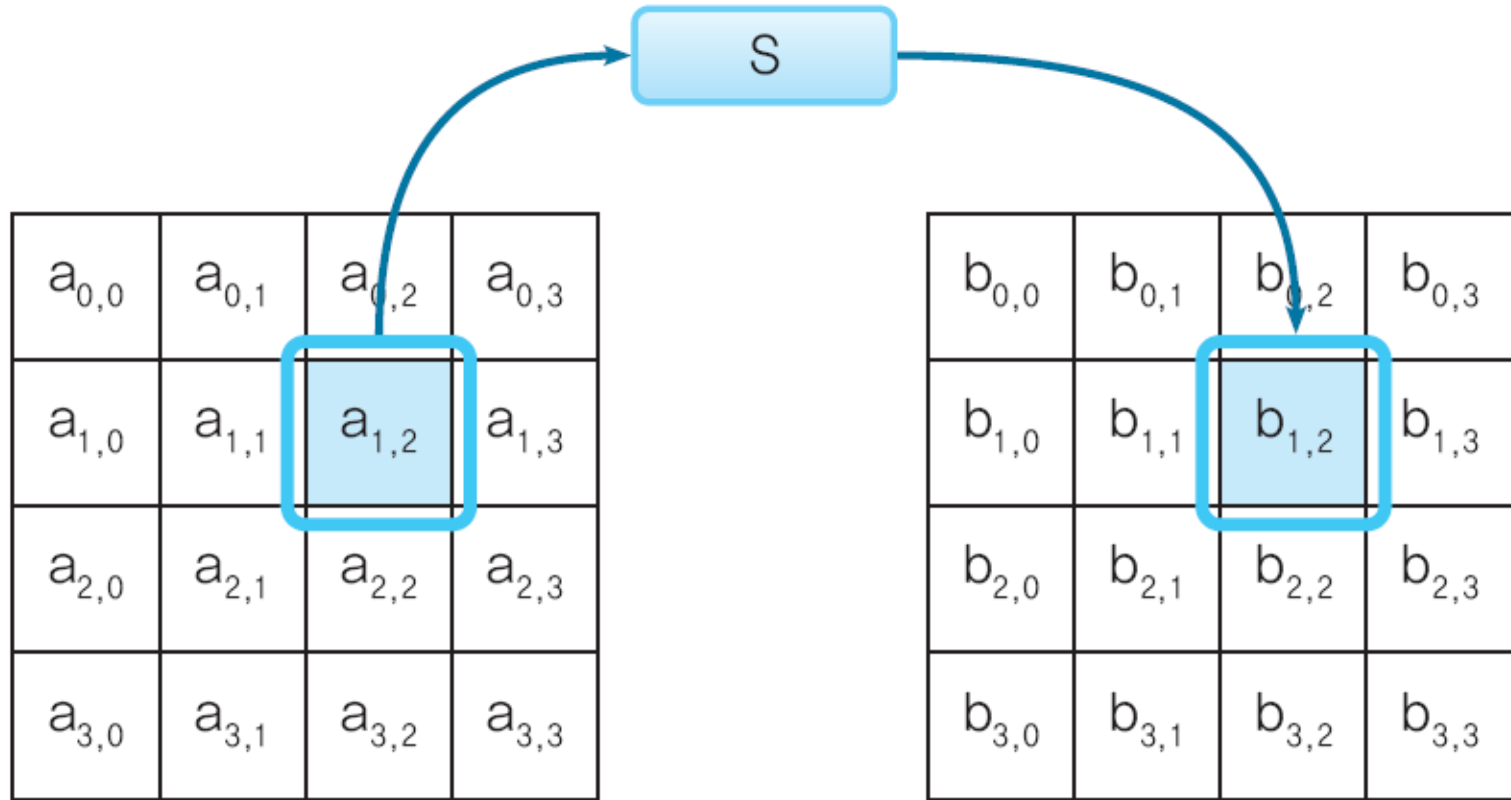


그림 4-11 • SubBytes(바이트 대체)

# ShiftRows(행 이동)

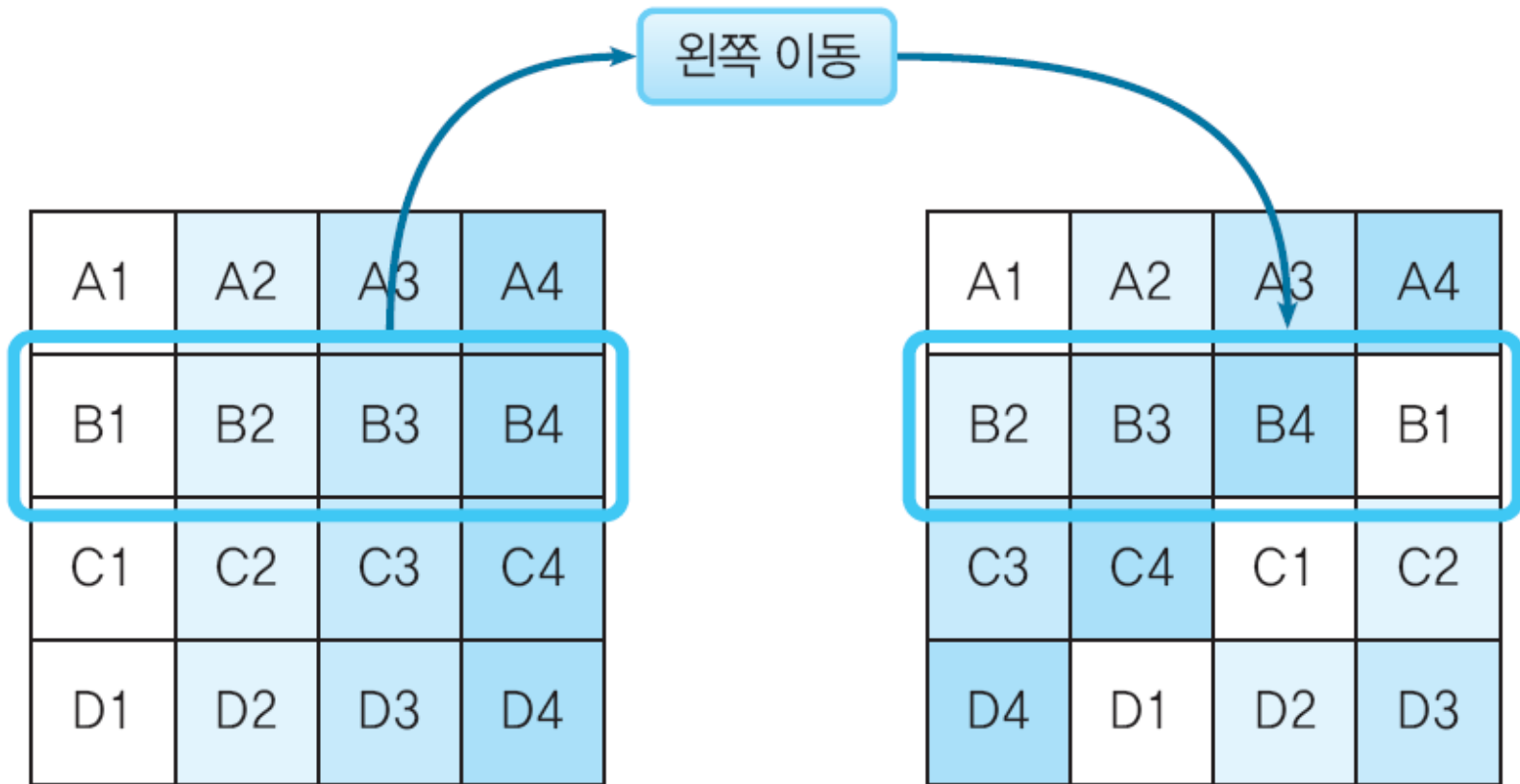


그림 4-12 • ShiftRows(행 이동)

# MixColumns(열 섞기)

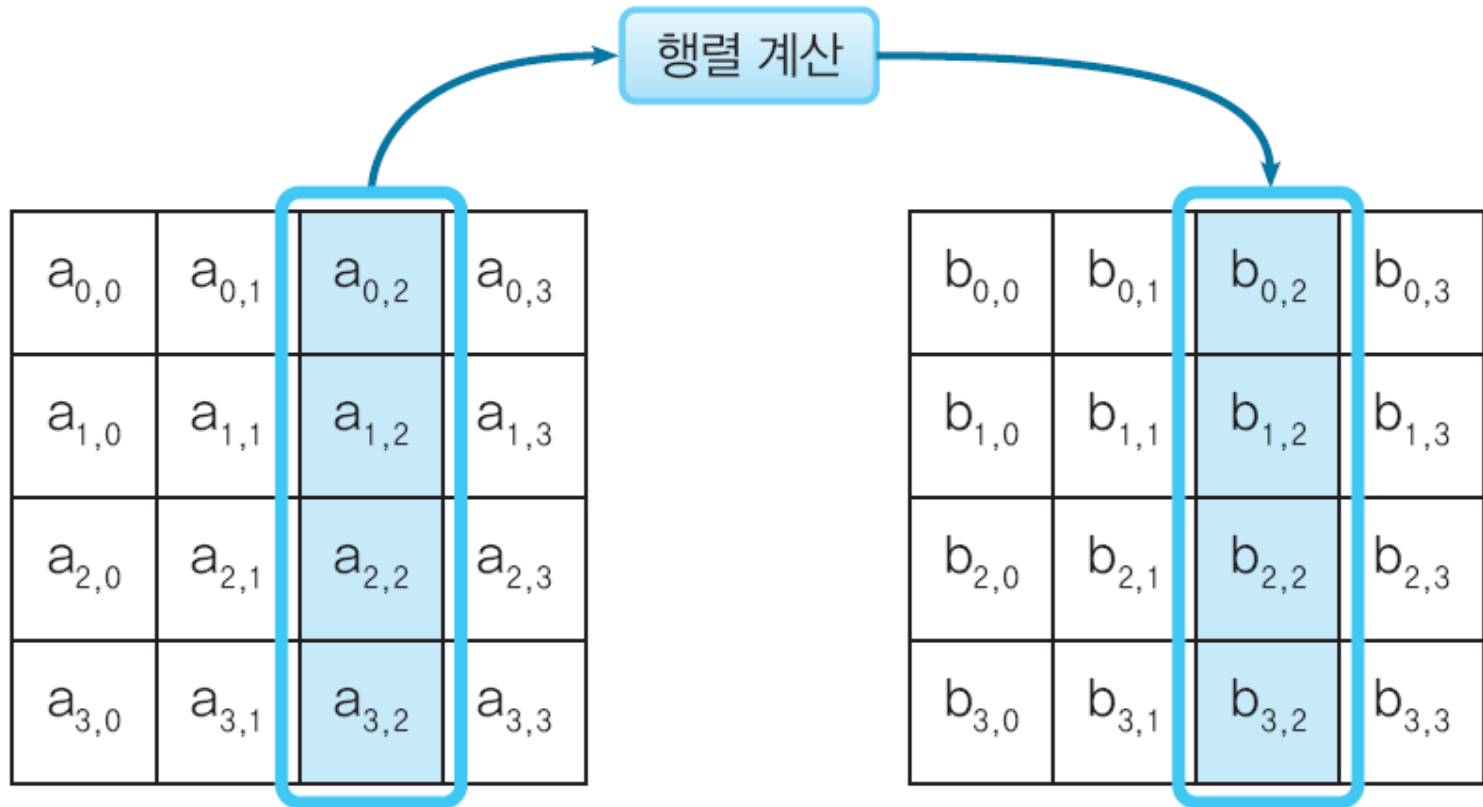


그림 4-13 • MixColumns(열 섞기)

# AddRoundKey(라운드 키와 XOR)

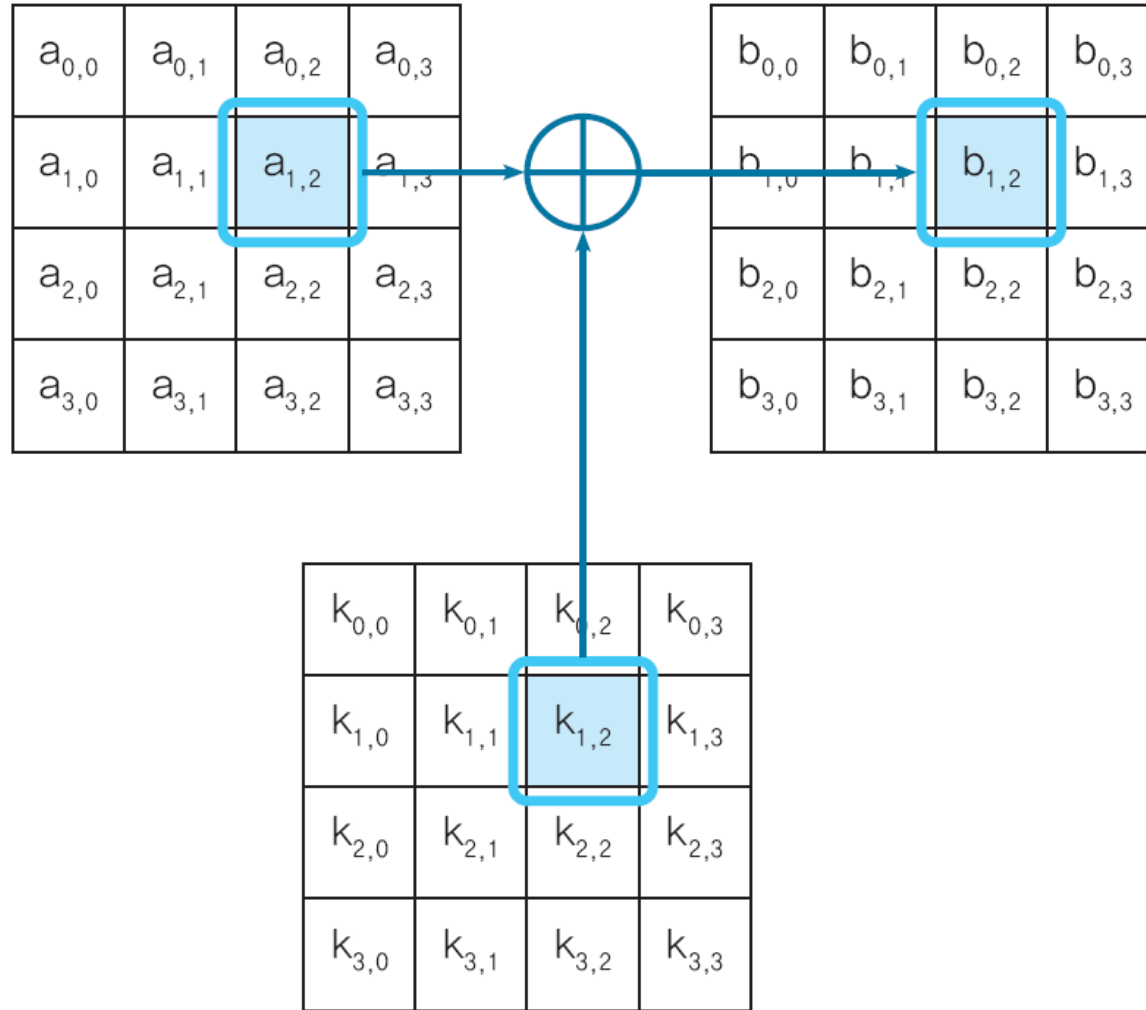


그림 4-14 • AddRoundKey(라운드 키와 XOR)

# InvMixColumns(역 열 섞기)

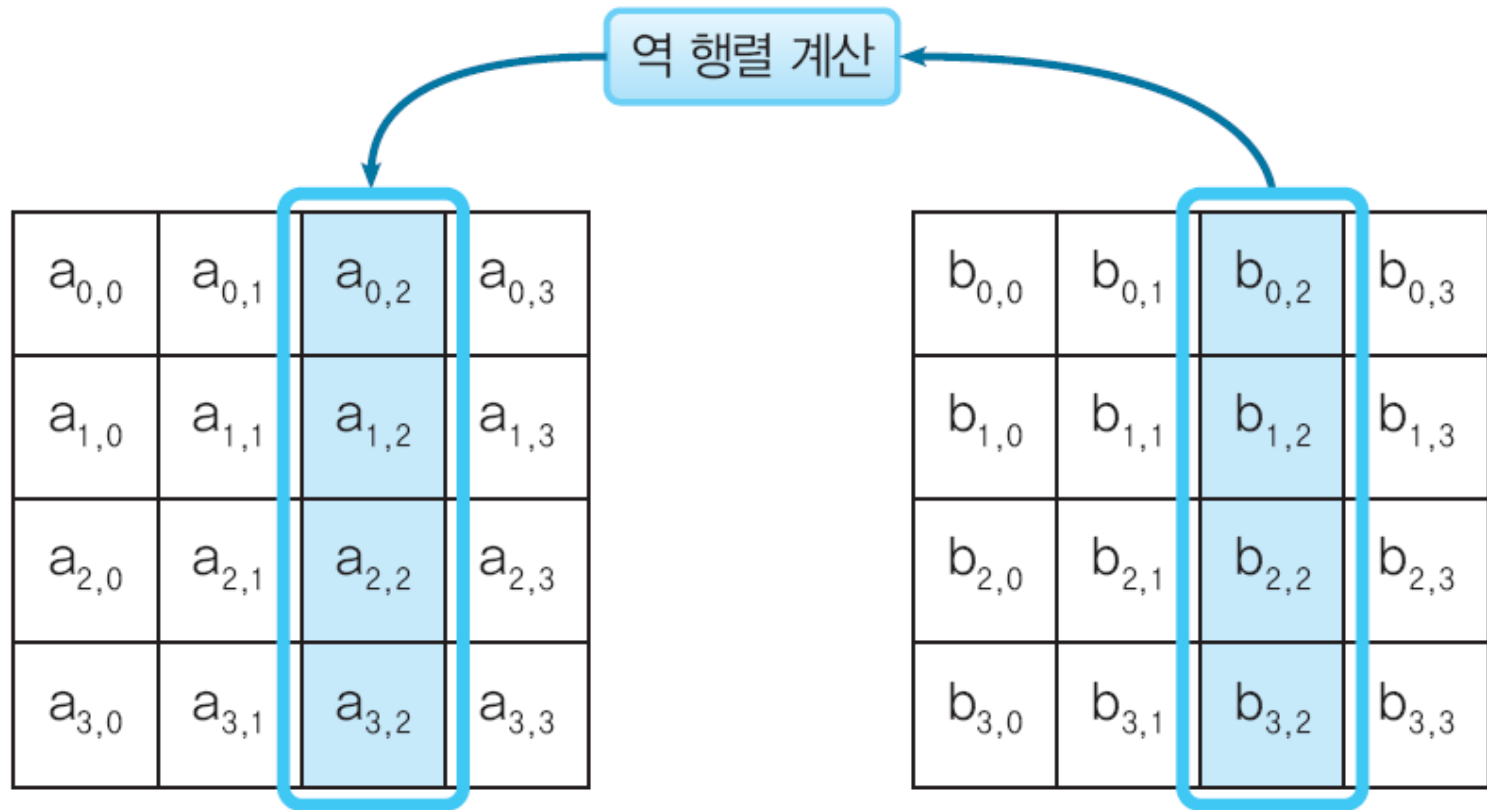


그림 4-15 • InvMixColumns(역 열 섞기)



# InvShiftRows(역 행 이동)

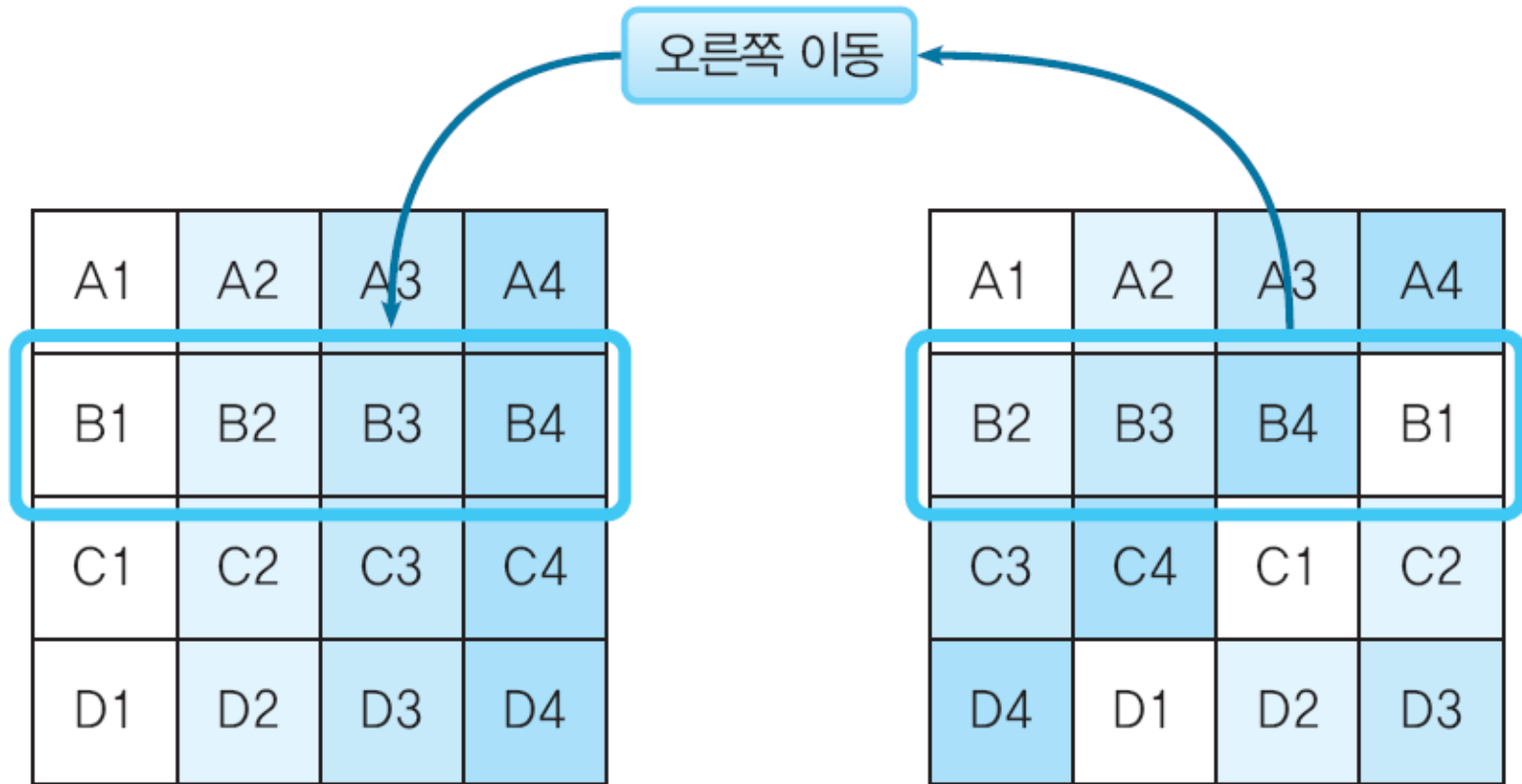


그림 4-16 • InvShiftRows(역 행 이동)

# InvSubBytes(역 바이트 대체)

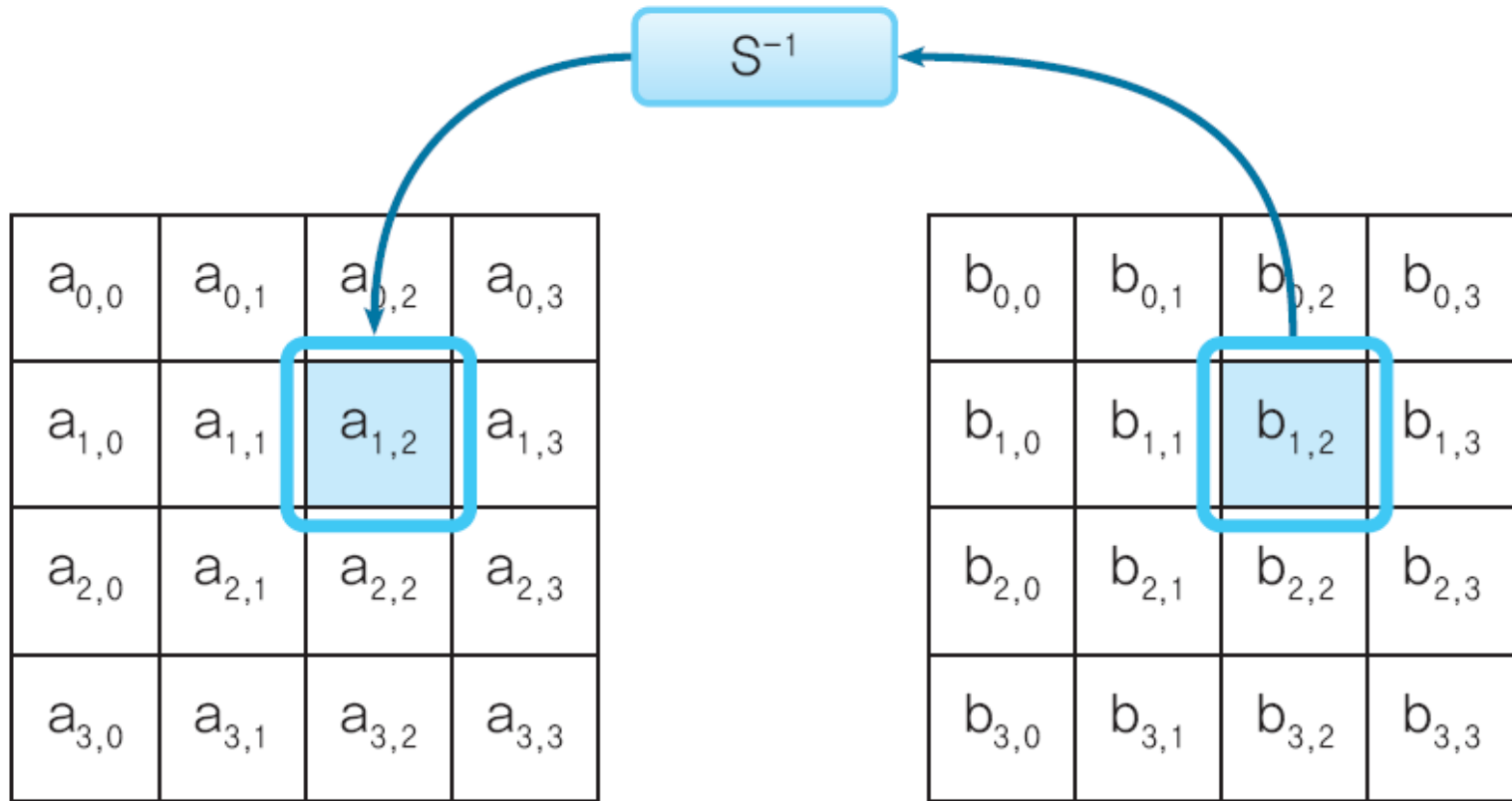


그림 4-17 • InvSubBytes(역 바이트 대체)

## 6.3 Rijndael의 해독

- Rijndael 알고리즘의 수학적 구조
  - Rijndael의 수식을 수학적인 조작에 의해 풀 수 있다면, Rijndael을 수학적으로 해독할 수 있을 것이다
- Rijndael에 대한 유효한 공격은 현재로서는 발견되지 않았다.

## 6.4 어떤 암호를 사용하면 좋은가?

- DES
  - 사용하지 말것
  - 과거 소프트웨어와의 호환성 유지를 위해 필요
- 트리플 DES
  - 호환성 때문에 앞으로도 당분간 사용
  - 점차 AES로 대체
- AES(Rijndael)
  - 고속
  - 다양한 플랫폼
  - 현재 까지 안전
  - 사용 권장
  - AES 최종 후보 5개도 사용가능

**Q & A**

**Thank You!**