알기 쉬운

정보보호개론 3판

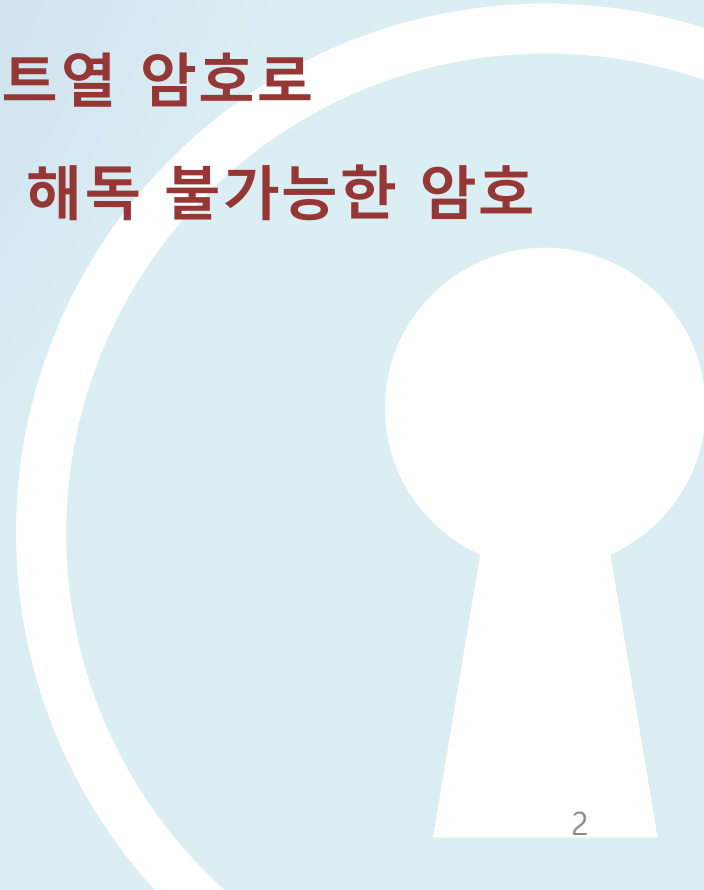
흥미로운 암호 기술의 세계

INFORMATION SECURITY and CRYPTOGRAPHY

INFINITYBOOKS
인피니티북스

INFORMATION SECURITY and CRYPTOGRAPHY

CHAPTER 4 대칭 암호



Section 01
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

Quiz 1 문자를 수로 대응시키는 것과 시저 암호

- 시저 암호에서 사용되는 알파벳은 A부터 Z까지 26문자이다. 지금 여기에서 A를 0, B를 1, …Z를 25라는 숫자에 대응시킨다. 이와 같이 문자를 수로 대응시킬 때 시저 암호에서 『3문자 앞으로 평행 이동』이라는 암호화는 어떻게 계산하는 것인가?

1.2 XOR

- XOR은‘익스클루시브 오아(exclusive or)’, 또는 짧게 ‘엑스오아’라고 읽는다.
- 우리 말로는 **배타적 논리합**

0 XOR 0 = 0 (0과 0의 XOR은 0이 된다)
0 XOR 1 = 1 (0과 1의 XOR은 1이 된다)
1 XOR 0 = 1 (1과 0의 XOR은 1이 된다)
1 XOR 1 = 0 (1과 1의 XOR은 0이 된다)

한 비트의 XOR

- XOR은 ⊕이라는 기호를 써서 표현

a ⊕ b	설명
0 ⊕ 0 = 0	0과 0의 XOR은 0이 된다
0 ⊕ 1 = 1	0과 1의 XOR은 1이 된다
1 ⊕ 0 = 1	1과 0의 XOR은 1이 된다
1 ⊕ 1 = 0	1과 1의 XOR은 0이 된다

- 같은 숫자끼리의 XOR은 반드시 0이 된다

0 ⊕ 0 = 0
1 ⊕ 1 = 0

비트열 XOR

0 1 0 0 1 1 0 0 … A

⊕ 1 0 1 0 1 0 1 0 … B

---

1 1 1 0 0 1 1 0 … A ⊕ B

비트열 XOR

1 1 1 0 0 1 1 0 … A ⊕ B

⊕ 1 0 1 0 1 0 1 0 … B

---

0 1 0 0 1 1 0 0 … A ⊕ B ⊕ B = A

(A로 돌아간다)

암호화/복호화의 순서와 매우 비슷

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

XOR은 그림을 마스크한다

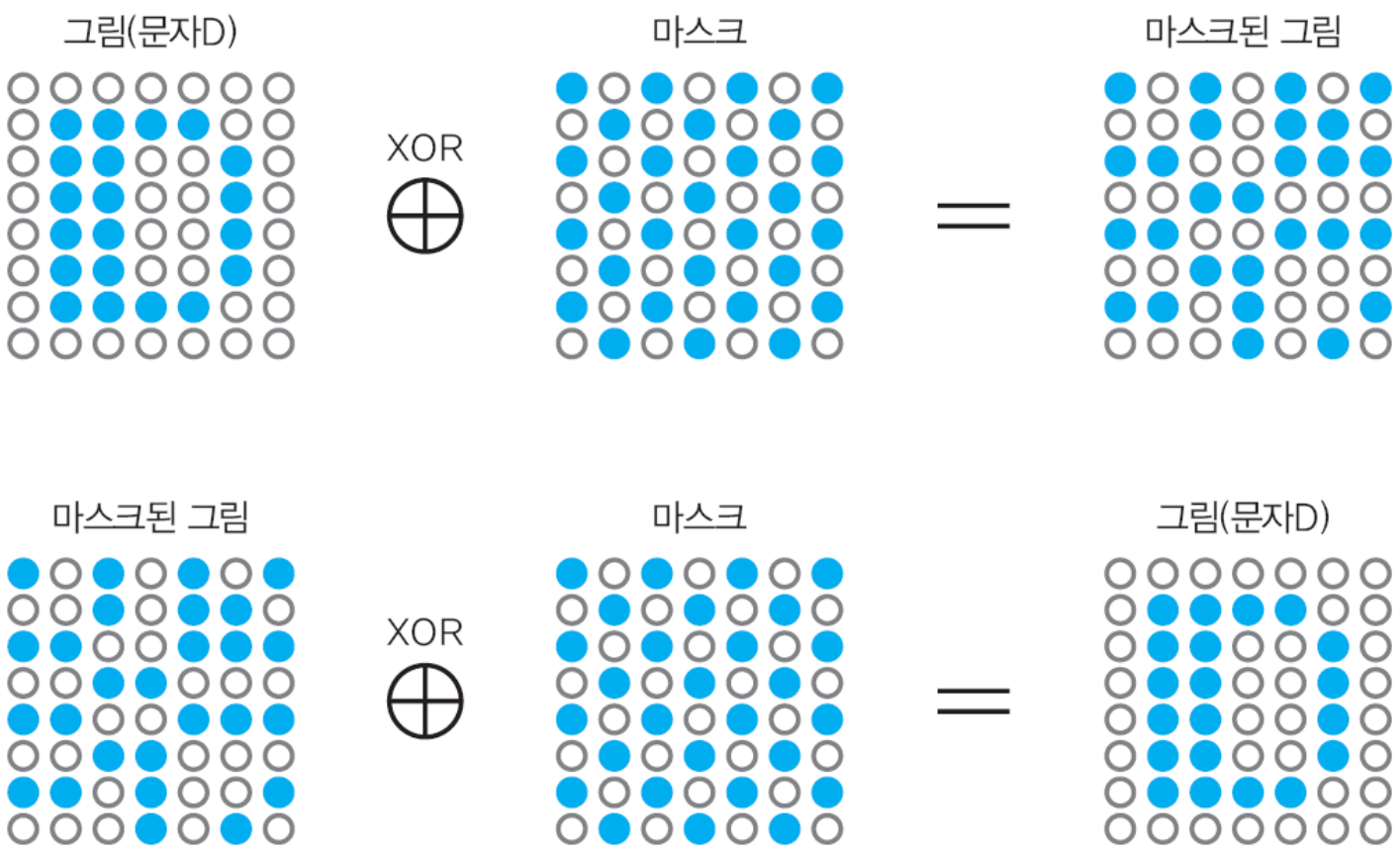


그림 4-1 XOR은 그림을 마스크한다

Section 02
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

2.5 일회용 패드는 왜 사용되지 않은 것일까?

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
⊕	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
⊕	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 복호화 해보면 도중에 모든 64비트 패턴이 등장한다
- 그 중에 나타날 수 있는 문자열 들
  - 규칙적인 문자열
    - aaaaaaaa, abcdefgh, zzzzzzzz 등
  - 의미 있는 영어 단어
    - midnight, onenight, mistress 등
  - 무의미한 문자열
    - %Ta_AjvX, HY(&JY!z, $@~*₩^^), Er#f6)(%
- 따라서 어느 것이 바른 평문인지 알 수 없다
  - 즉 어떤 키를 사용하면 바르게 복호화 할 수 있는지 알 수 없다

전사 공격

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

2.5 일회용 패드는 왜 사용되지 않은 것일까?

- 키 배송
  - 키의 길이가 통신문의 길이와 같다
  - 키를 안전하게 보낼 수 있는 방법이 있다면 평문 그 자체를 같은 방법으로 안전하게 보낼 수 있다
- 키 보존
  - 평문과 같은 비트 길이의 키를 안전하게 보존할 수 있다면 평문 그 자체를 안전하게 보존해 둘 수 있다

2.5 일회용 패드는 왜 사용되지 않은 것일까?

- 키 재이용
  - 과거에 사용한 랜덤한 비트열을 절대로 재이용해서는 안 된다
- 키 동기화
  - 통신하는 동안 송신자와 수신자 사이에 키가 되는 비트열이 1 비트라도 어긋나서는 안 된다
- 키 생성
  - 난수를 대량으로 생성할 필요가 있다
  - 난수는 의사 난수가 아니라 실제 난수이어야 한다

Quiz 2 일회용 패드와 압축

일회용 패드의 이야기를 들은 앨리스는 다음과 같이 생각했다.

일회용 패드에서는 키의 길이가 평문의 길이와 같다고 하는데, 나는 데이터를 압축하는 프로그램을 가지고 있다. 이것을 사용하면 일회용 패드의 키를 압축해서 짧게 할 수 있지 않을까?

앨리스의 생각은 옳은가?

Section 03
DES

- DES(Data Encryption Standard)는 1977년에 미국의 연방 정보처리 표준 규격(FIPS)으로 채택된 대칭 암호
- 전사공격으로 해독할 수 있는 수준

DES 콘테스트(DES Challenge)

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

3.1 암호화/복호화

- DES(Data Encryption Standard) : 64비트 평문을 64비트 암호문으로 암호화하는 대칭 암호 알고리즘
- 키의 비트 길이는 56비트 : 실제로 64비트이지만 7비트마다 오류 검출 정보 1비트 추가됨
- 64비트 평문(비트열)을 하나의 단위로 모아서 암호화(블럭암호)

블록 암호

- **블록 암호**(block cipher)
  - 블록 단위로 처리를 하는 암호 알고리즘
  - 긴 비트 길이의 평문을 암호화하기 위해서는 평문을 64비트 블록으로 나누고 각각을 DES로 반복하여 암호화함
  - 반복하여 암호화 하는 것을 mode라 함

DES의 암호화 · 복호화

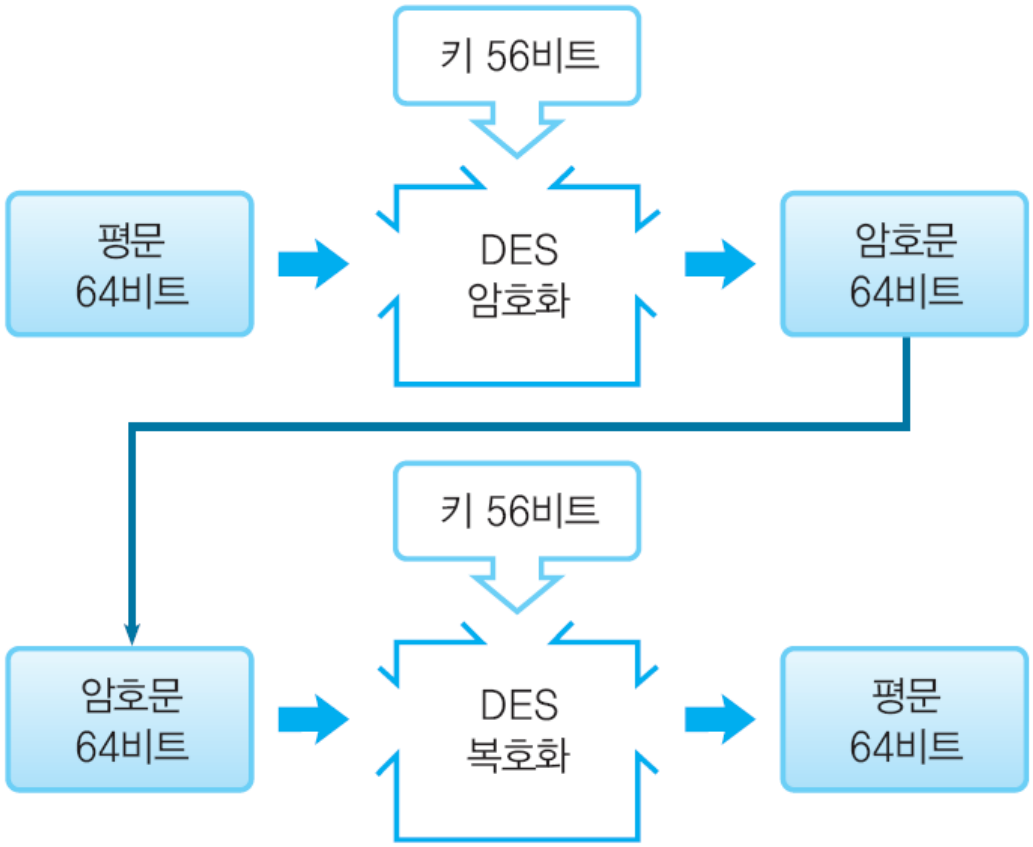


그림 4-2 DES의 암호화 · 복호화

DES 구조

- 페이스텔 네트워크(Feistel network)
  - 페이스텔 구조(Feistel structure)
  - 페이스텔 암호(Feistel cipher)
  - DES 외의 다른 블록암호도 채용
  - 여러 개의 라운드(round)로 구성
  - DES는 16라운드로 구성

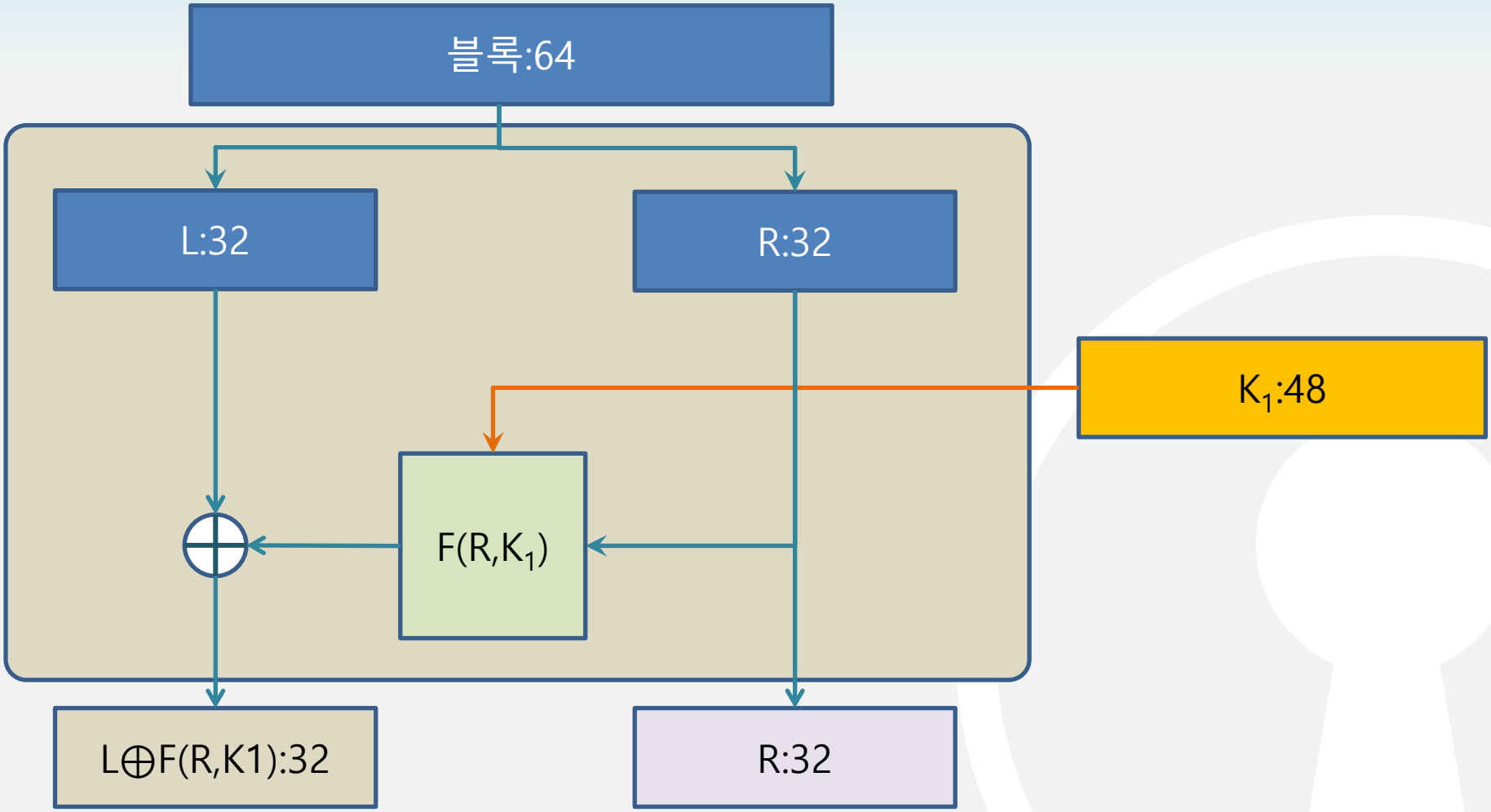
페이스텔 네트워크

- 용어
  - 평문의 왼쪽 반 L, 오른쪽 반 R
  - 서브키(Subkey) : 해당 라운드에서 부분적으로 사용하는 키
  - 라운드(round) 함수 F : R과 서브키를 가지고 L을 암호화 하기 위한 비트열을 생성하는 함수

페이스텔 네트워크의 1 라운드

1. 입력을 L과 R로 나눈다
2. R을 그대로 R로 보낸다
3. R을 라운드 함수 F로 보낸다
4. 라운드 함수 F는 R과 서브 키 K_1을 입력으로 사용하여 랜덤하게 보이는 비트열을 계산한다
5. 얻어진 비트열과 L을 XOR 한다
6. 그 결과를 다음 라운드의 L로 사용한다

페이스텔 네트워크 1 라운드



페이스텔 네트워크의 1 라운드

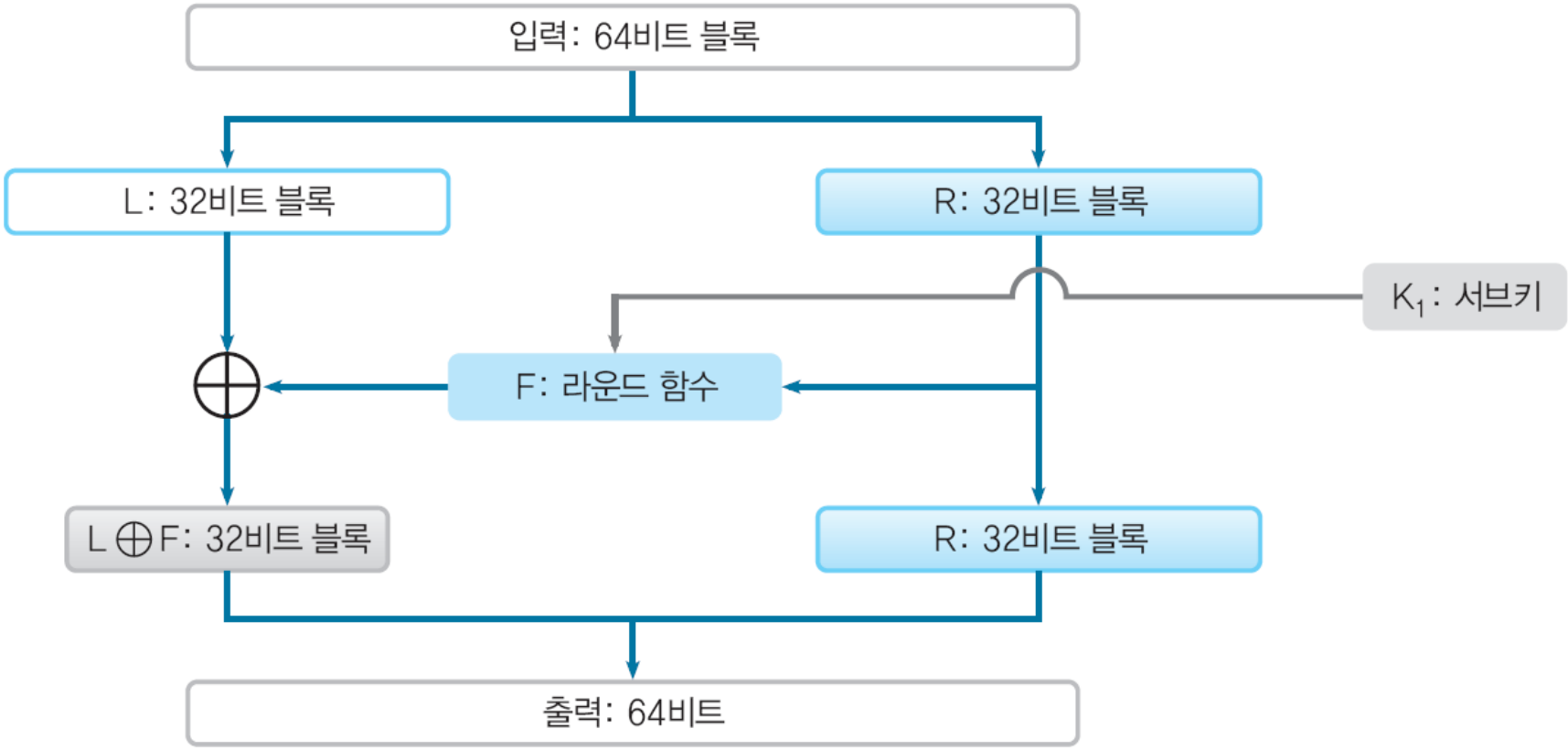
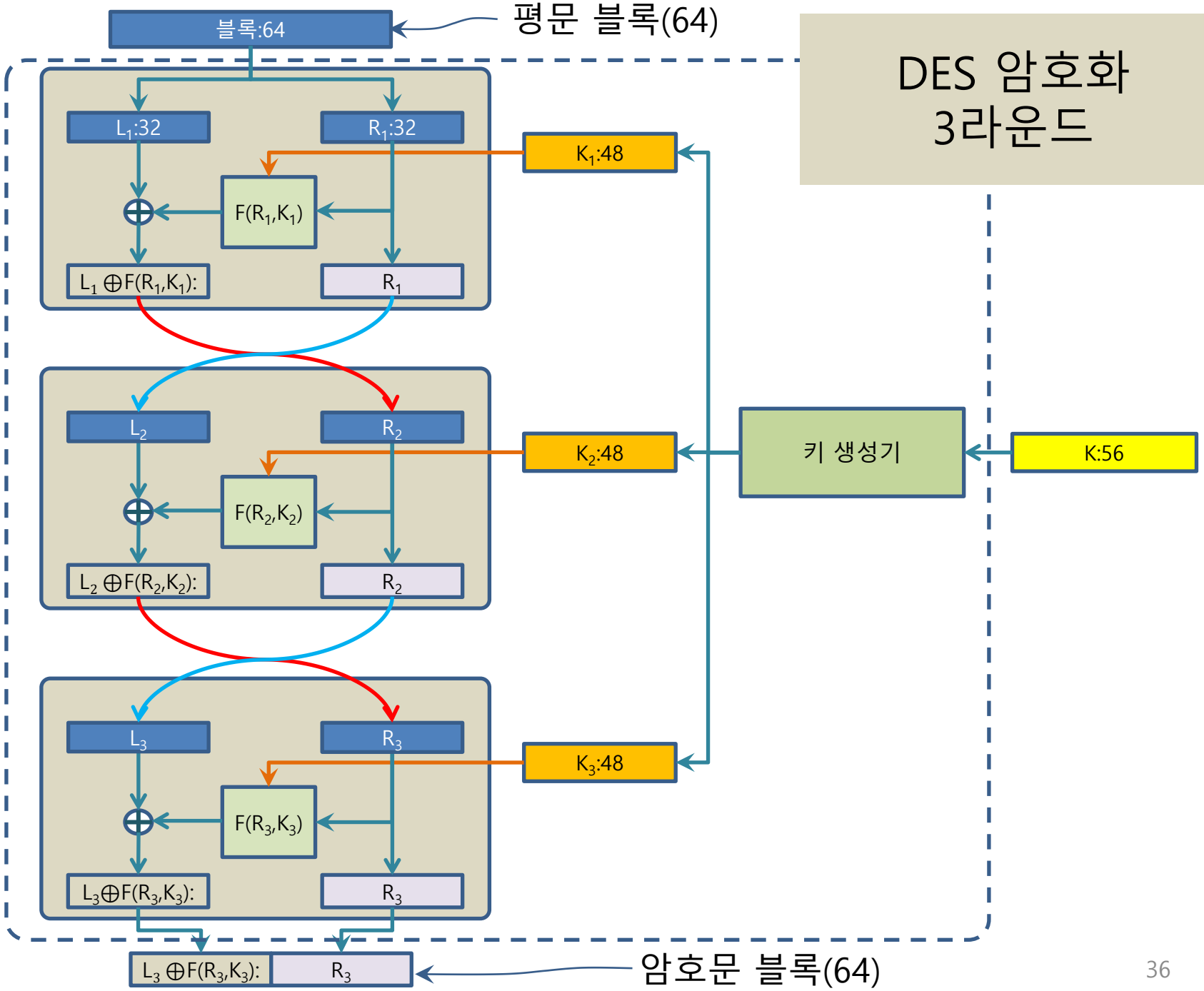


그림 4-3 페이스텔 네트워크의 1 라운드

페이스텔 네트워크의 1 라운드

- 1라운드 후 R은 전혀 암호화 되지 않음
- 다음 라운드에서 L과 R을 교환하여 입력
- 서브키는 새로운 서브키를 사용


평문 블록(64)
DES 암호화
3라운드
블록:64
L_1:32
R_1:32
K_1:48
$F(R_1,K_1)$
$L_1 \oplus F(R_1,K_1)$:
R_1
L_2
R_2
K_2:48
$F(R_2,K_2)$
$L_2 \oplus F(R_2,K_2)$:
R_2
키 생성기
K:56
L_3
R_3
K_3:48
$F(R_3,K_3)$
$L_3 \oplus F(R_3,K_3)$:
R_3
$L_3 \oplus F(R_3,K_3)$:
R_3
암호문 블록(64)

페이스텔 네트워크 3 라운드

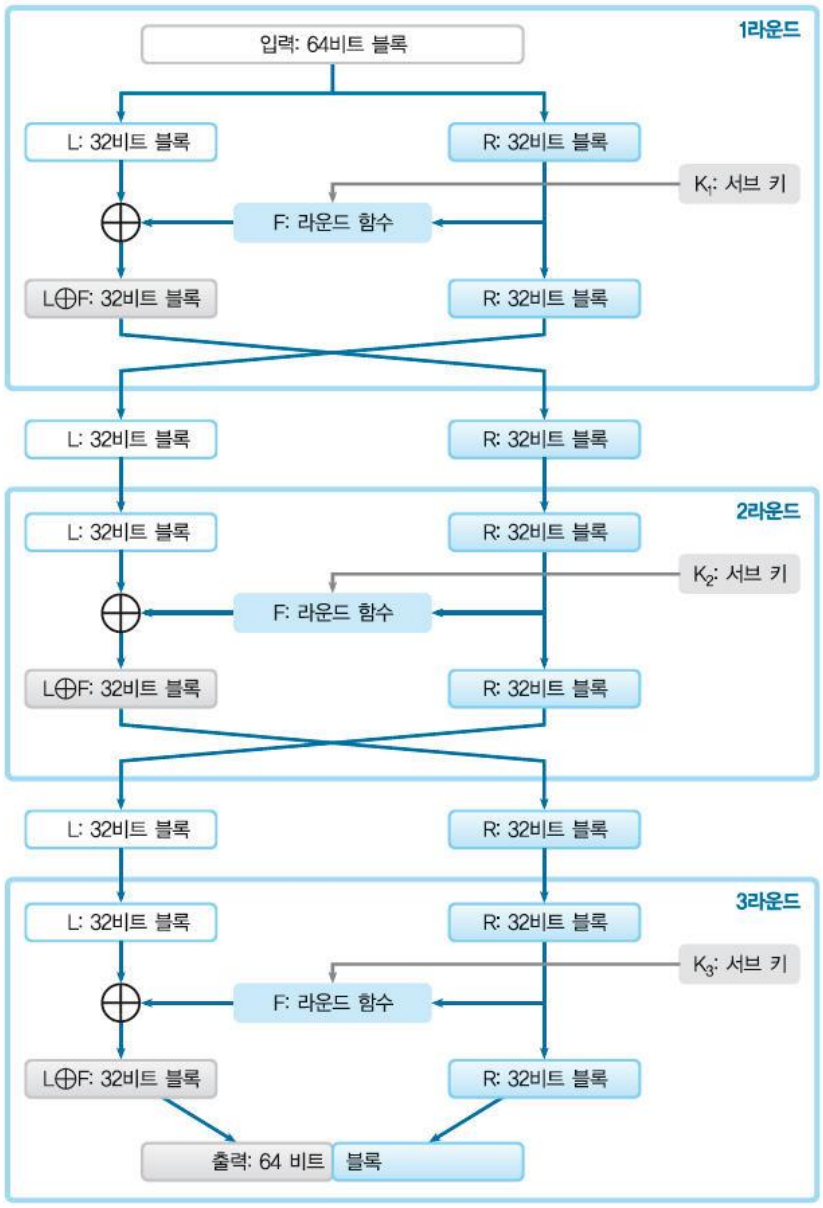
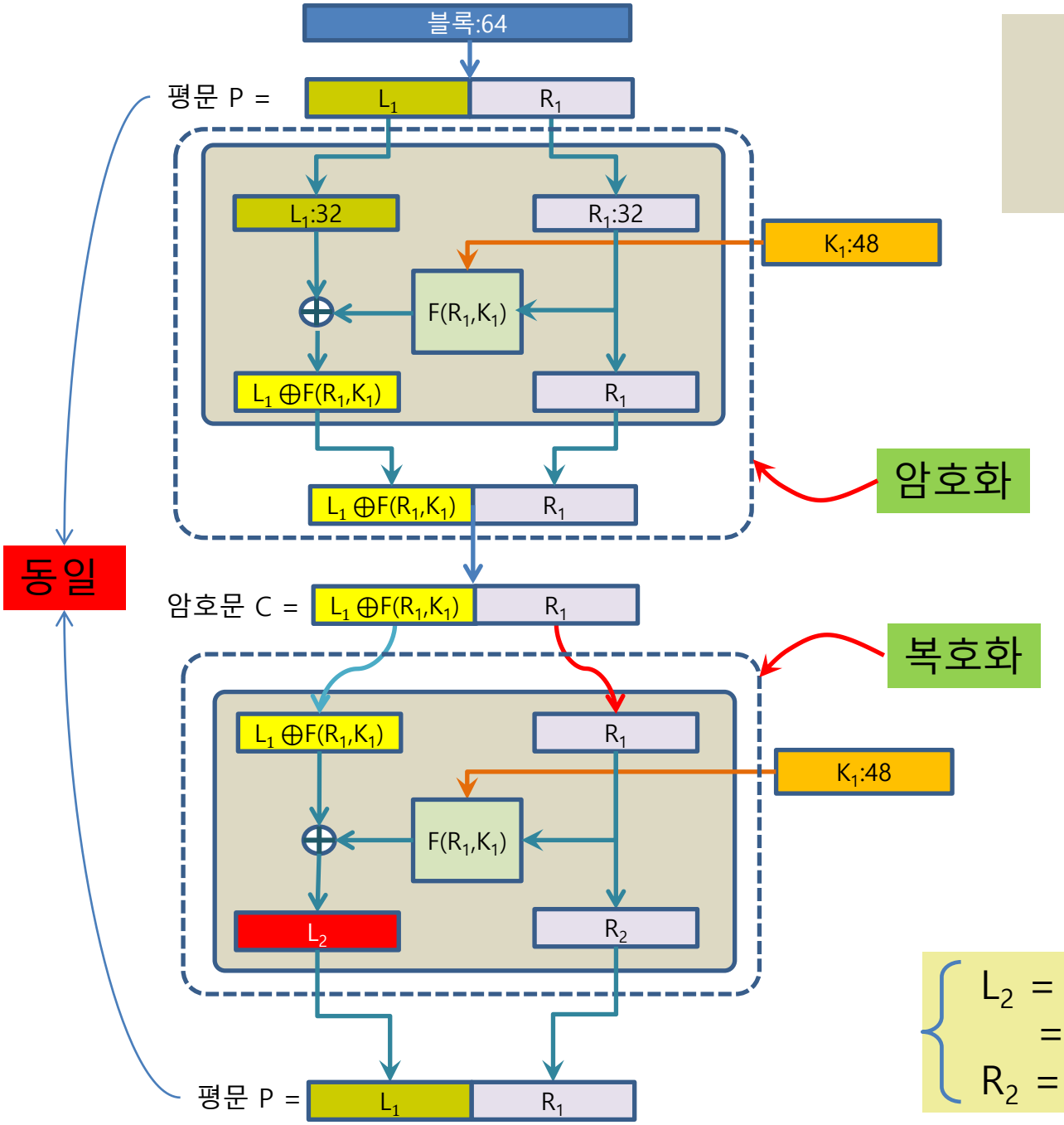


그림 4-4 • 페이스텔 네트워크의 암호화(3라운드)

DES 2회 통과



$$\begin{cases} L_2 = [L_1 \oplus F(R_1,K_1)] \oplus F(R_1,K_1) \\ \quad\;\; = L_1 \\ R_2 = R_1 \end{cases}$$

페이스텔 네트워크 2회 통과

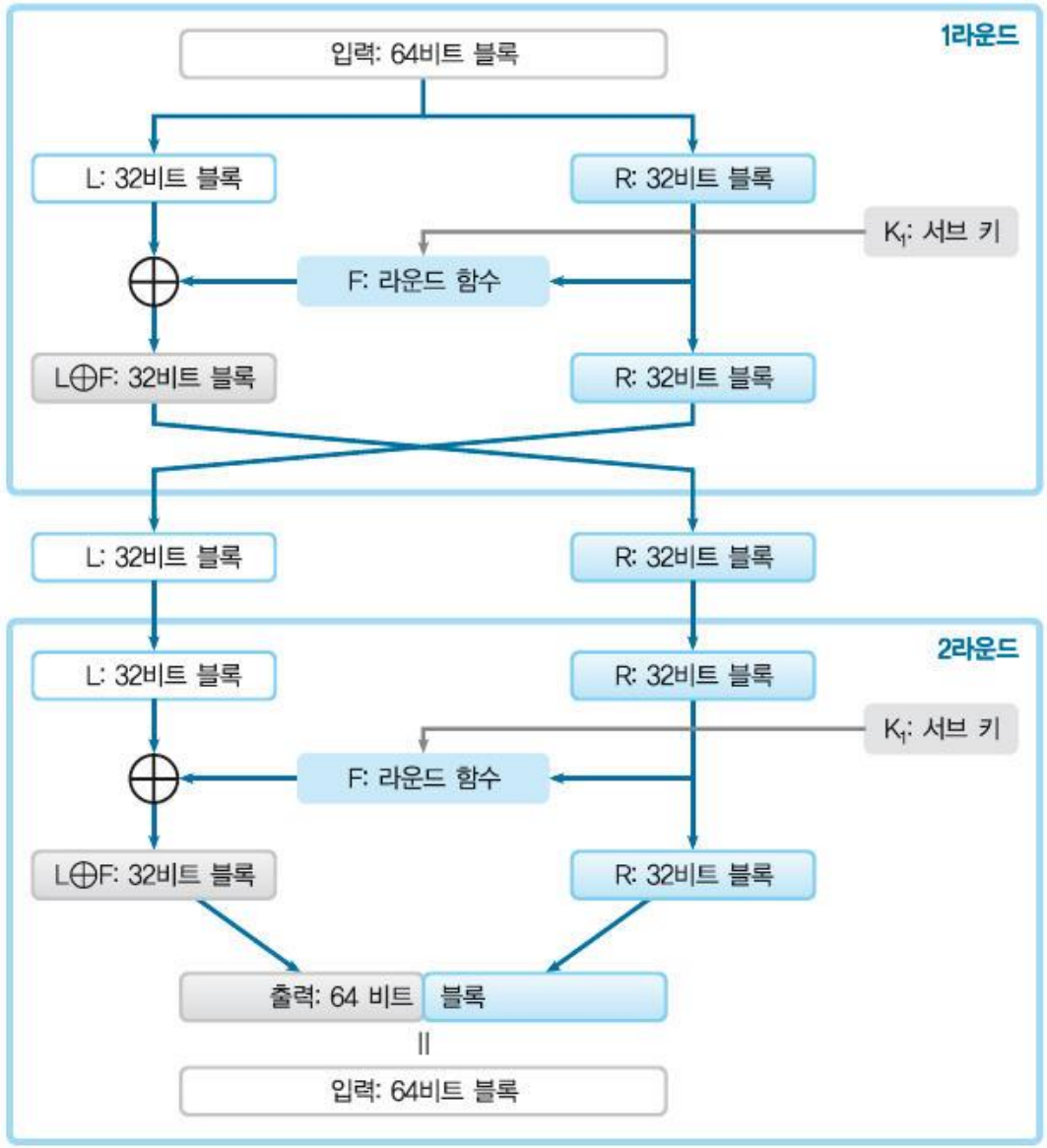
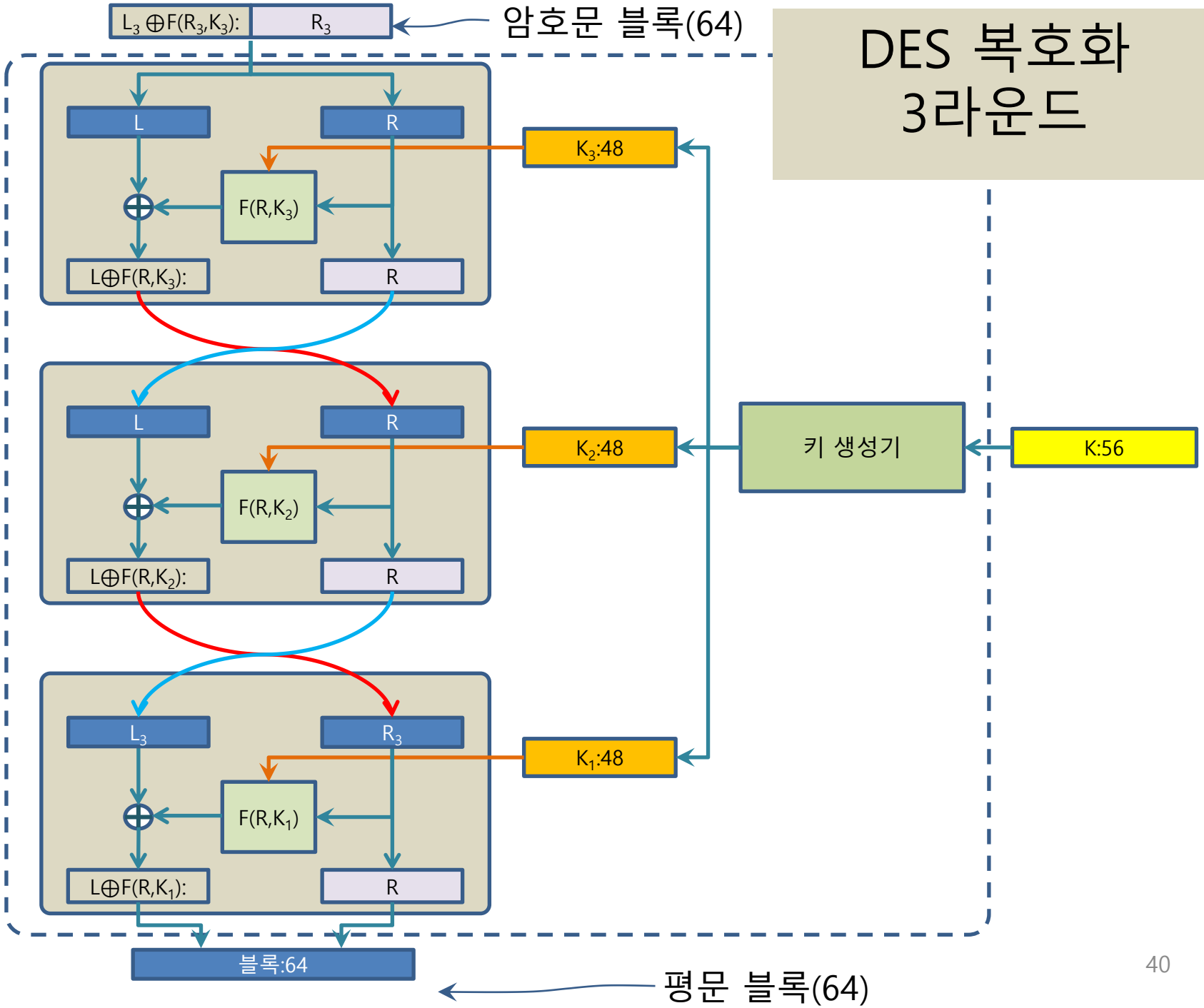


그림 4-5 • 같은 서브 키로 페이스텔 네트워크를 2회 통과시키면 원래로 돌아간다


$L_3 \oplus F(R_3,K_3)$:
R_3
암호문 블록(64)
DES 복호화
3라운드
L
R
K_3:48
$F(R,K_3)$
$L \oplus F(R,K_3)$:
R
L
R
K_2:48
키 생성기
K:56
$F(R,K_2)$
$L \oplus F(R,K_2)$:
R
L_3
R_3
K_1:48
$F(R,K_1)$
$L \oplus F(R,K_1)$:
R
블록:64
평문 블록(64)

페이스텔 네트워크 복호화 (3라운드)

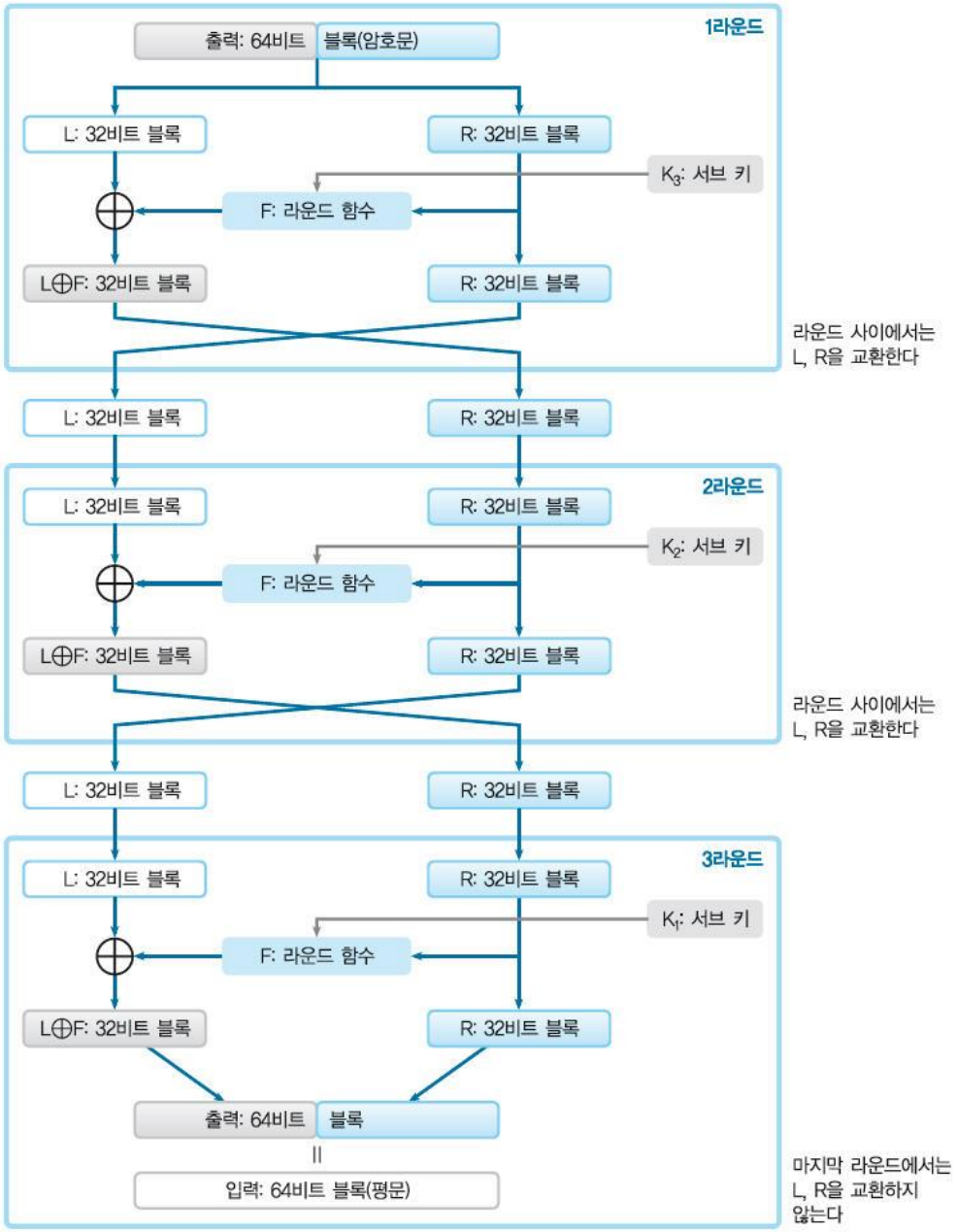


그림 4-6 • 페이스텔 네트워크의 복호화(3라운드)

페이스텔 네트워크의 특징

- 원하는 만큼 라운드수를 늘릴 수 있다
  - 아무리 라운드를 늘려도 복호화할 수 없게 될 염려가 없음
- 라운드 함수 F에 어떤 함수를 사용해도 복호화가 가능하다
  - 어떤 함수를 사용하더라도 복호화할 수 없게 될 염려가 없음
- 암호화와 복호화를 완전히 동일한 구조로 실현할 수 있다

3.2 차분 해독법과 선형 해독법

- **차분 해독법** (Differential Cryptanalysis)
  - 블록암호 해독법
  - Biham과 Shamir가 개발
  - 평문의 일부를 변경할 때 암호문이 어떻게 변화하는지 관찰하여 조사하는 암호 해독법
  - 입력하는 평문의 한 비트라도 달라지면 암호문이 달라지는데 암호문의 변화의 틀을 조사해서 해독하는 방법
- **선형 해독법** (Linear cryptanalysis)
  - 마츠이(Matsui)가 개발
  - 평문과 암호문 비트를 몇 개 정도 XOR 해서 0이 되는 확률을 조사하는 암호 해독법

차분/선형 해독법 전제 조건

- 가정 : 암호 해독자가 임의로 만든 평문을 암호화 할 수 있어야 한다
- 공격자는 **선택평문공격**(CPA: Chosen Plaintext Attack)을 할 수 있다

AES(Adnvanced Encryption Standard)와 차분/선형 해독법

- AES(Advanced Encryption Standard)는 차분 해독법이나 선형 해독법으로 부터 안전하다

Section 04
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 가능
- DES를 대신할 블록 암호가 필요
- 이를 위해 개발된 것이 트리플 DES : TDES, 3DES
- DES보다 강력하도록 DES를 3단 겹치게 한 암호 알고리즘
- 트리플 DES의 키 : 56비트x3=168비트

4.2 트리플 DES 암호화

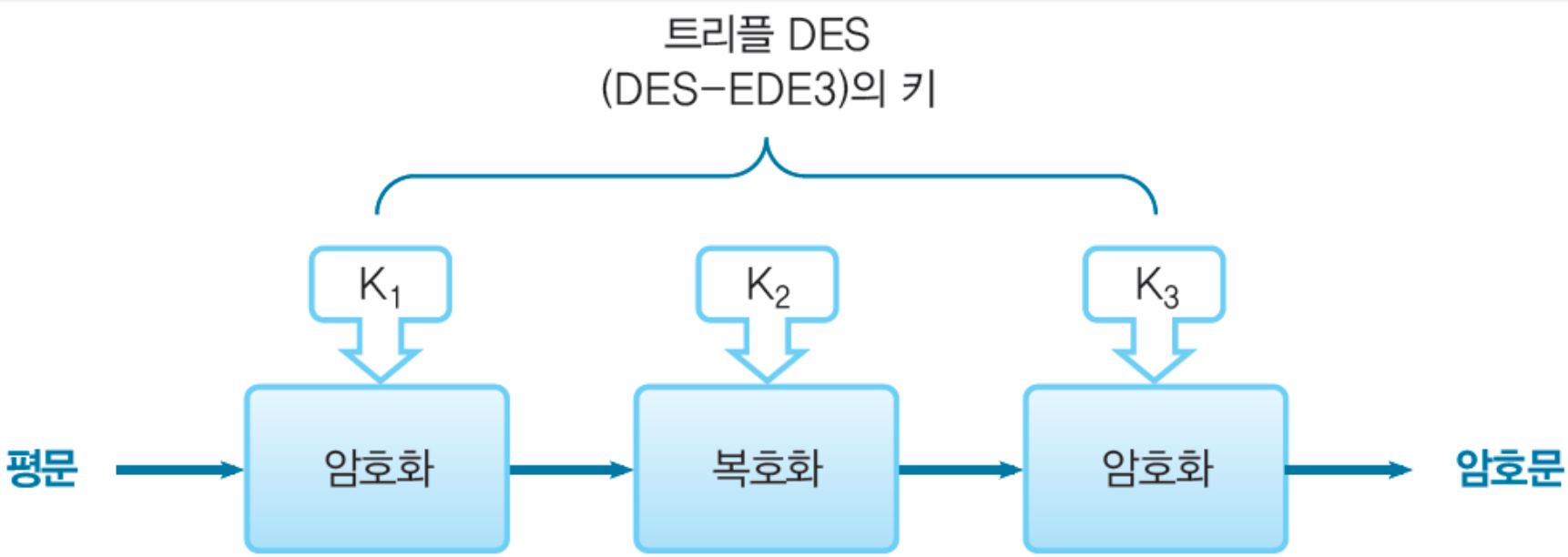


그림 4-7 트리플 DES의 암호화

트리플 DES는 DES로도 사용

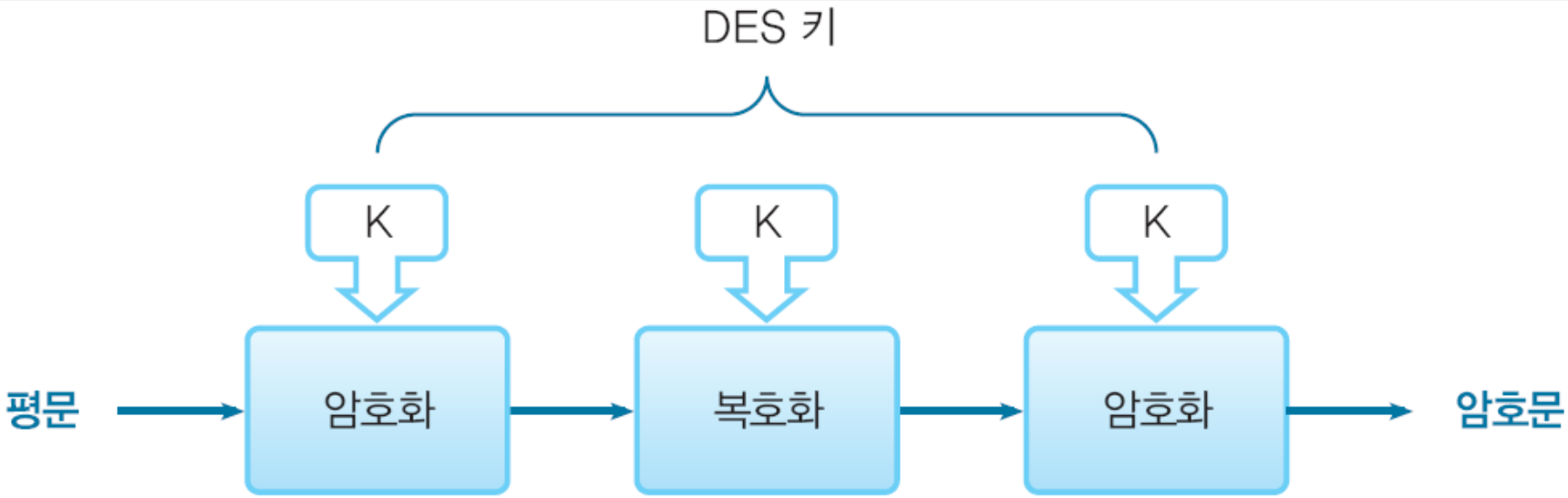


그림 4-8 트리플 DES는 DES로도 사용할 수 있다

트리플 DES 종류

- 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

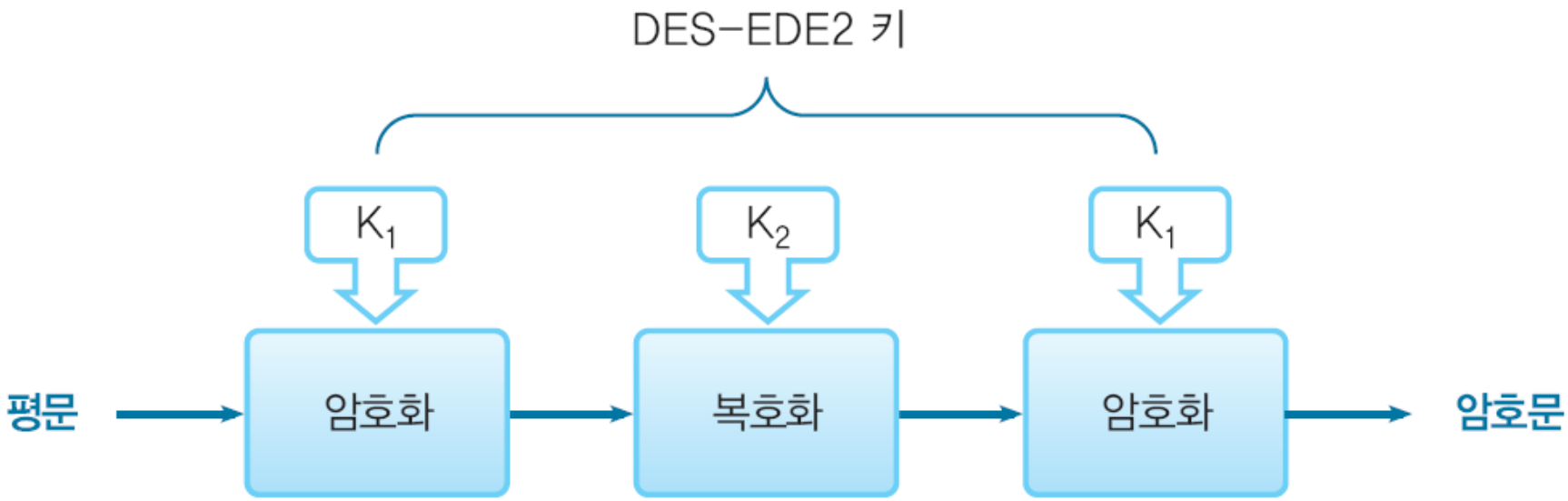


그림 4-9 DES-EDE2

4.3 트리플 DES 복호화

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

트리플 DES(DES-EDE3)의 복호화

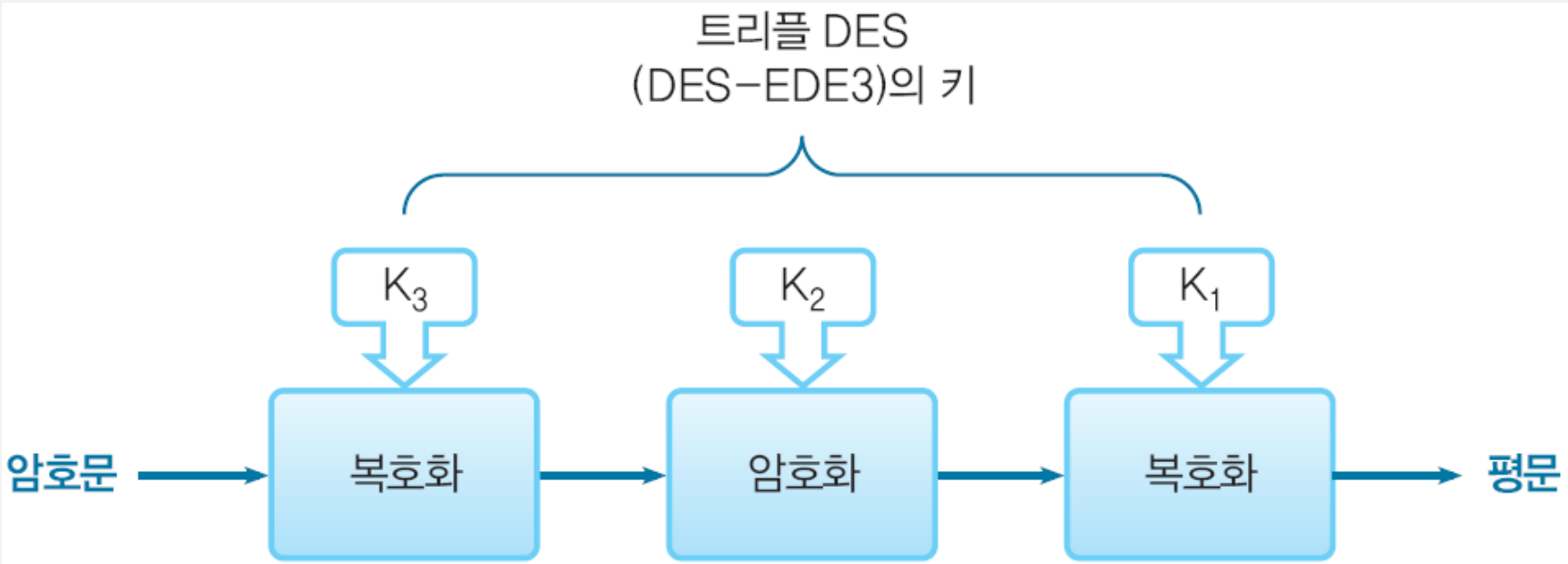


그림 4-10 트리플DES(DES-EDE3)의 복호화

4.4 트리플 DES 현황

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

Section 05
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(I)

- 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

5.2 AES 선정 과정(II)

- 선정 시 고려 조건
  - 약점의 유무
  - 속도가 빠를 것
  - 단순하고 구현하기 쉬울 것
  - 키의 설정 속도
  - 계산 능력이 낮은 플랫폼에서 고 성능의 플랫폼까지 모두 효율적으로 동작할 것

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	Counterpane사

- 최종 선정 : Rijndael(Joan Daemen, Vincent Rijmen)

Section 06
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 ~ 256비트
  - 32비트 단위로 선택
  - 실재로 128, 192, 256비트

6.2 Rijndael의 암호화와 복호화(I)

- 복수의 **라운드(round)**로 구성(10~14)
- **SPN(Substitution-Permutation Network)** 구조
- **SubBytes(바이트 대체)**
- **ShiftRows(행 이동)**
- **MixColumns(열 섞기)**
- **AddRoundKey(라운드 키와 XOR)**

6.2 Rijndael의 암호화와 복호화(II)

- AES 규격 : 입력 블록 128비트, 키 길이 128, 192, 256 비트
- SubBytes (바이트 대체)
  - 16바이트 입력에 대하여 각각 1바이트씩 처리
  - 1바이트 값을 인덱스하고 256개의 치환표로부터 1개의 값을 얻음
  - 256 문자판과 단일치환암호와 동일

6.2 Rijndael의 암호화와 복호화(II)

- ShiftRow (행 이동)
- MixColumns(열 섞기)
  - 4바이트의 값을 비트 연산을 써서 다른 4바이트 값으로 변환
- AddRoundKey(라운드키와 XOR)

SubByte(바이트 대체)

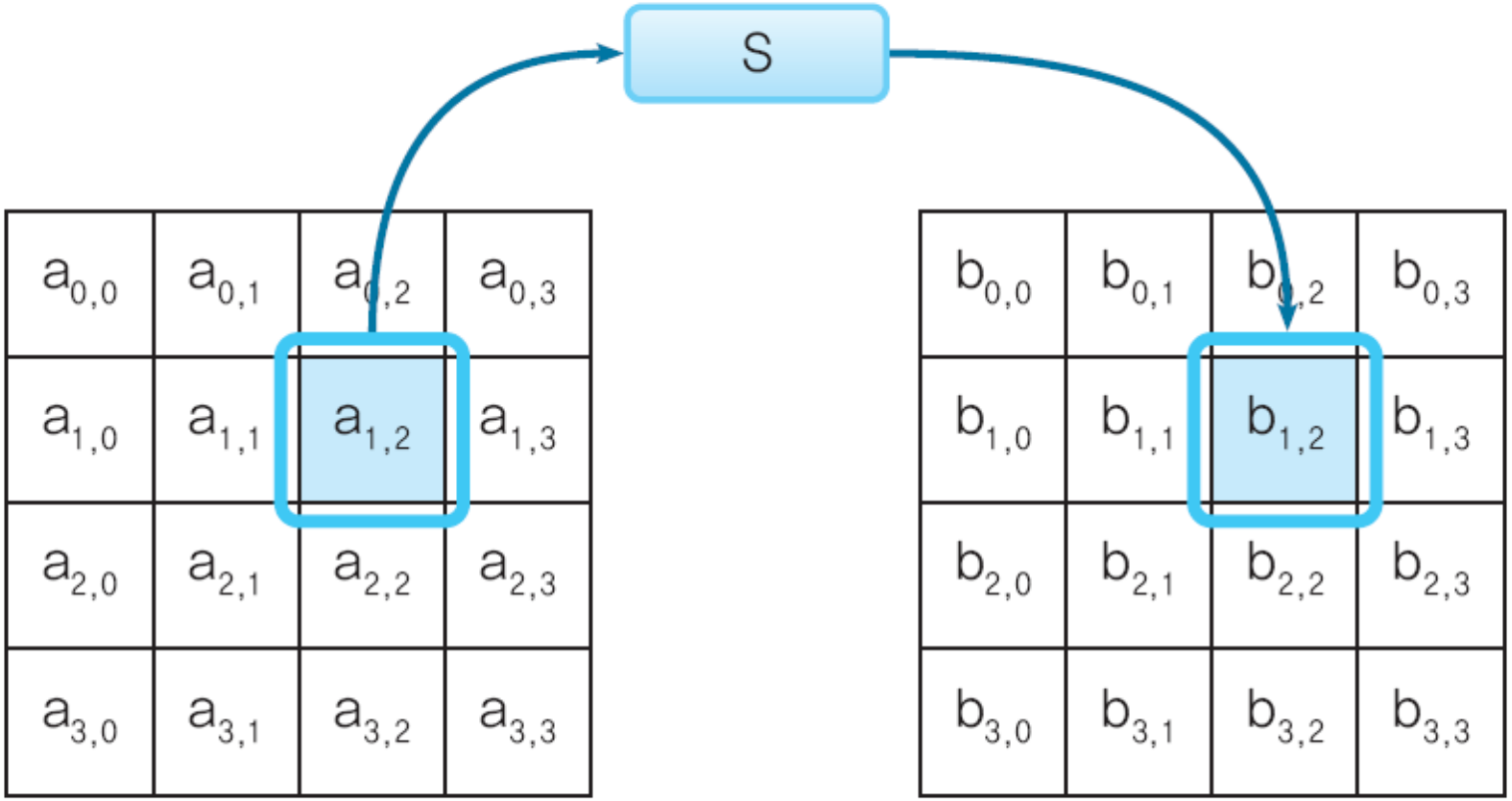


그림 4-11 • SubBytes(바이트 대체)

ShiftRows(행 이동)

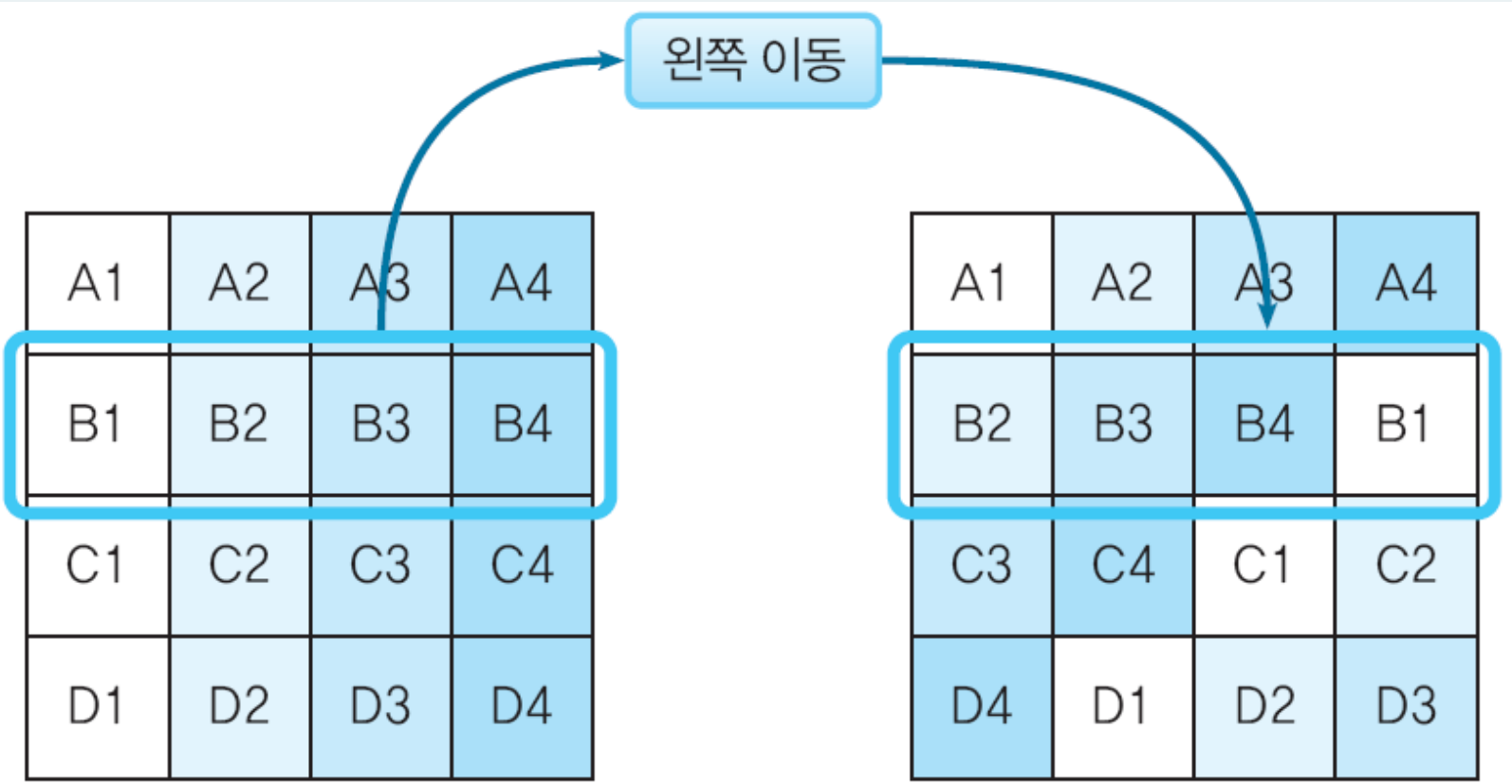


그림 4-12 • ShiftRows(행 이동)

MixColumns(열 섞기)

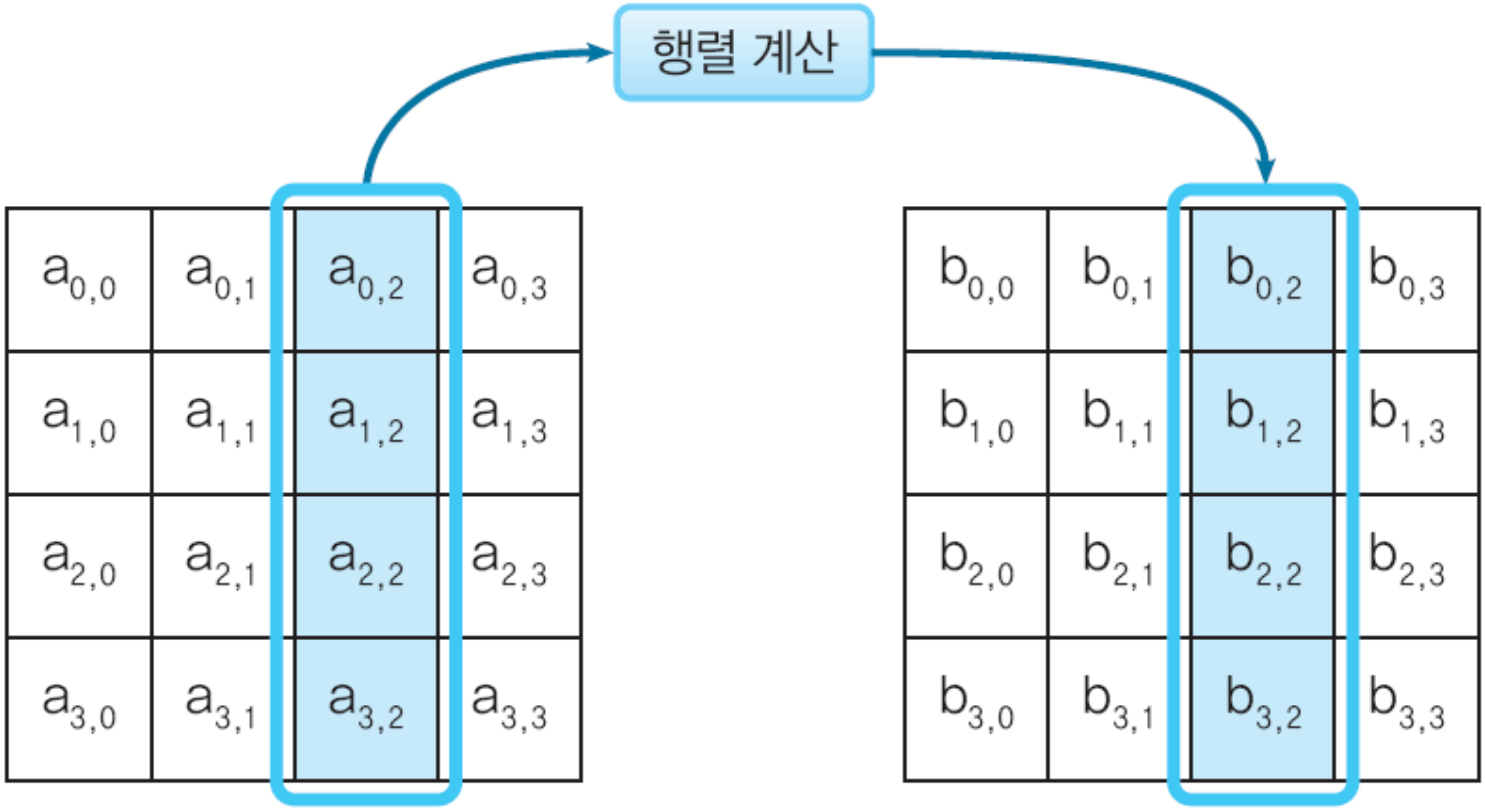


그림 4-13 • MixColumns(열 섞기)

AddRoundKey(라운드 키와 XOR)

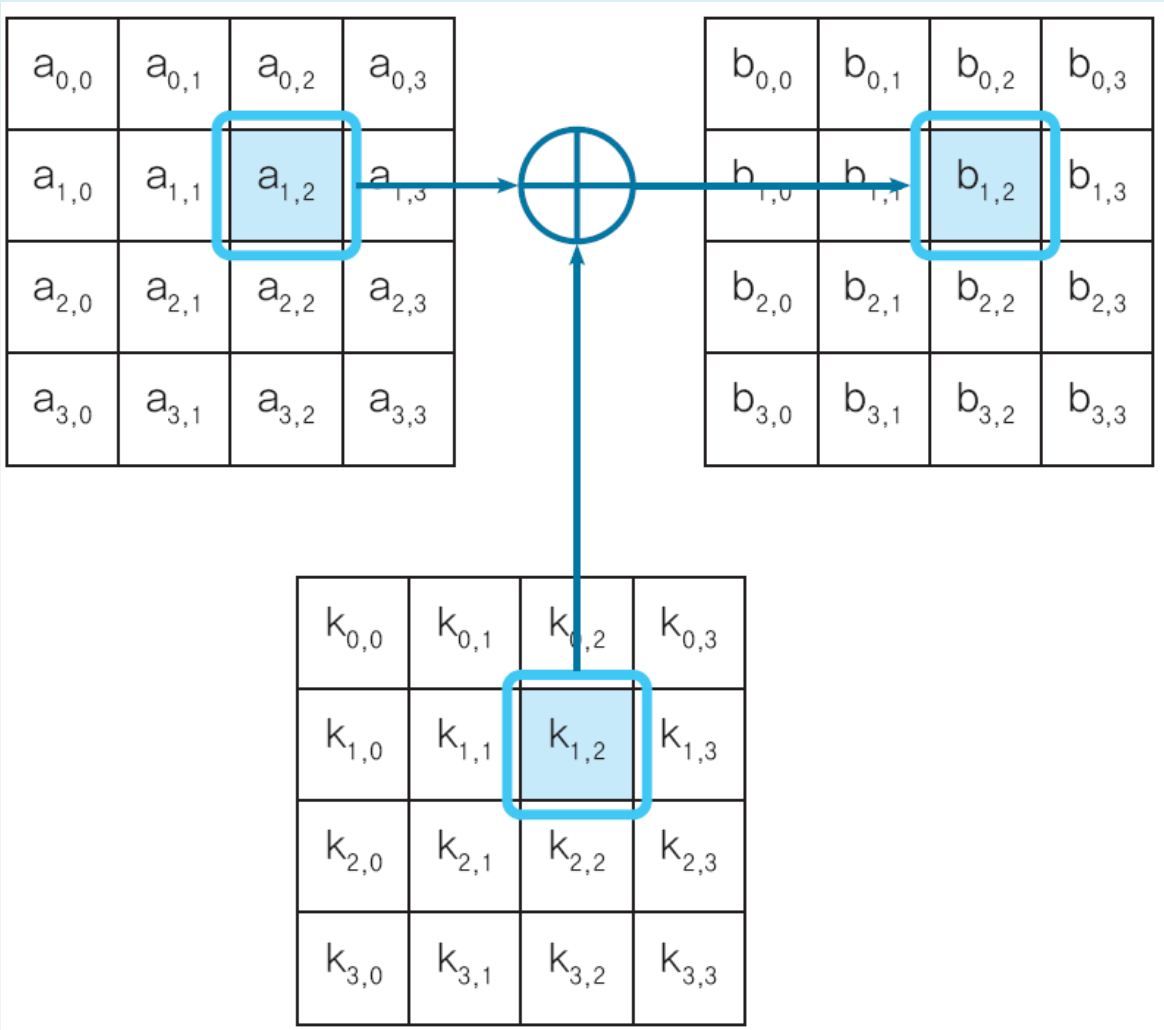


그림 4-14 • AddRoundKey(라운드 키와 XOR)

Rijndael의 복호화

- AddRoundKey->InvMixColumns->InvShiftRow ->InvSubBytes
- AddRoundKey에서 XOR 연산을 하므로 암호화와 복호화는 동일한 처리를 함

InvMixColumns(역 열 섞기)

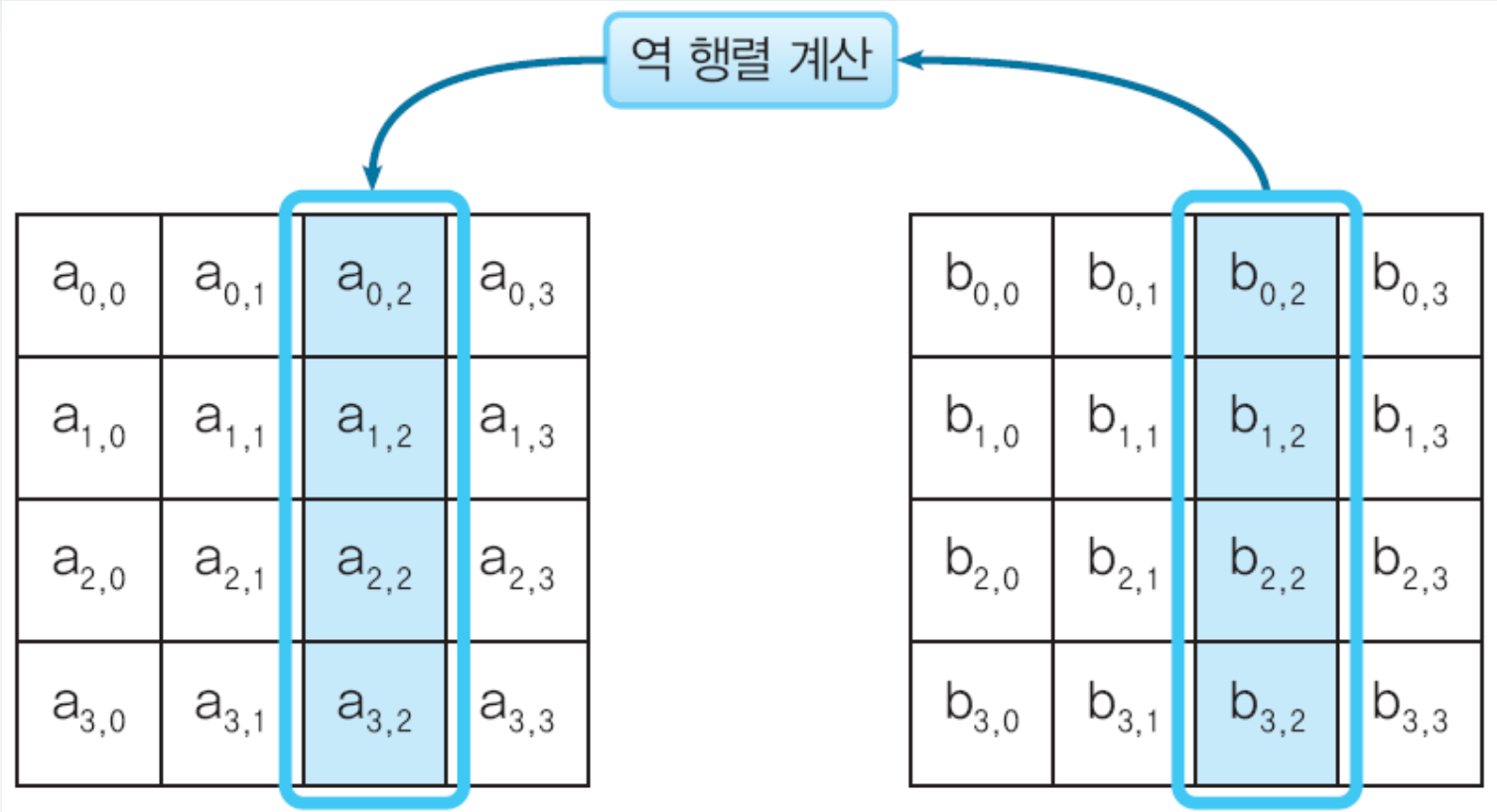


그림 4-15 • InvMixColumns(엮 열 섞기)

InvShiftRows(역 행 이동)

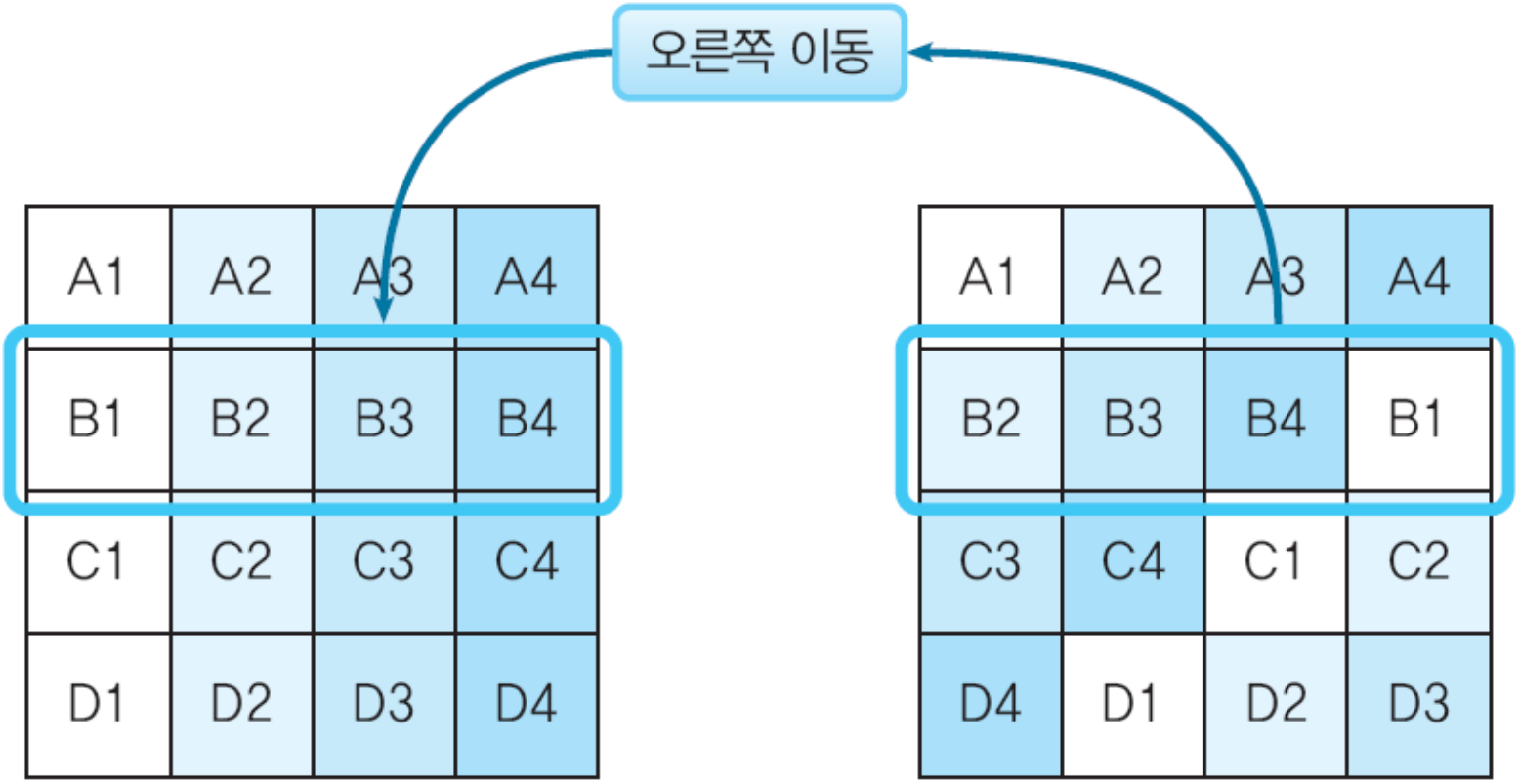


그림 4-16 • InvShiftRows(역 행 이동)

InvSubBytes(역 바이트 대체)

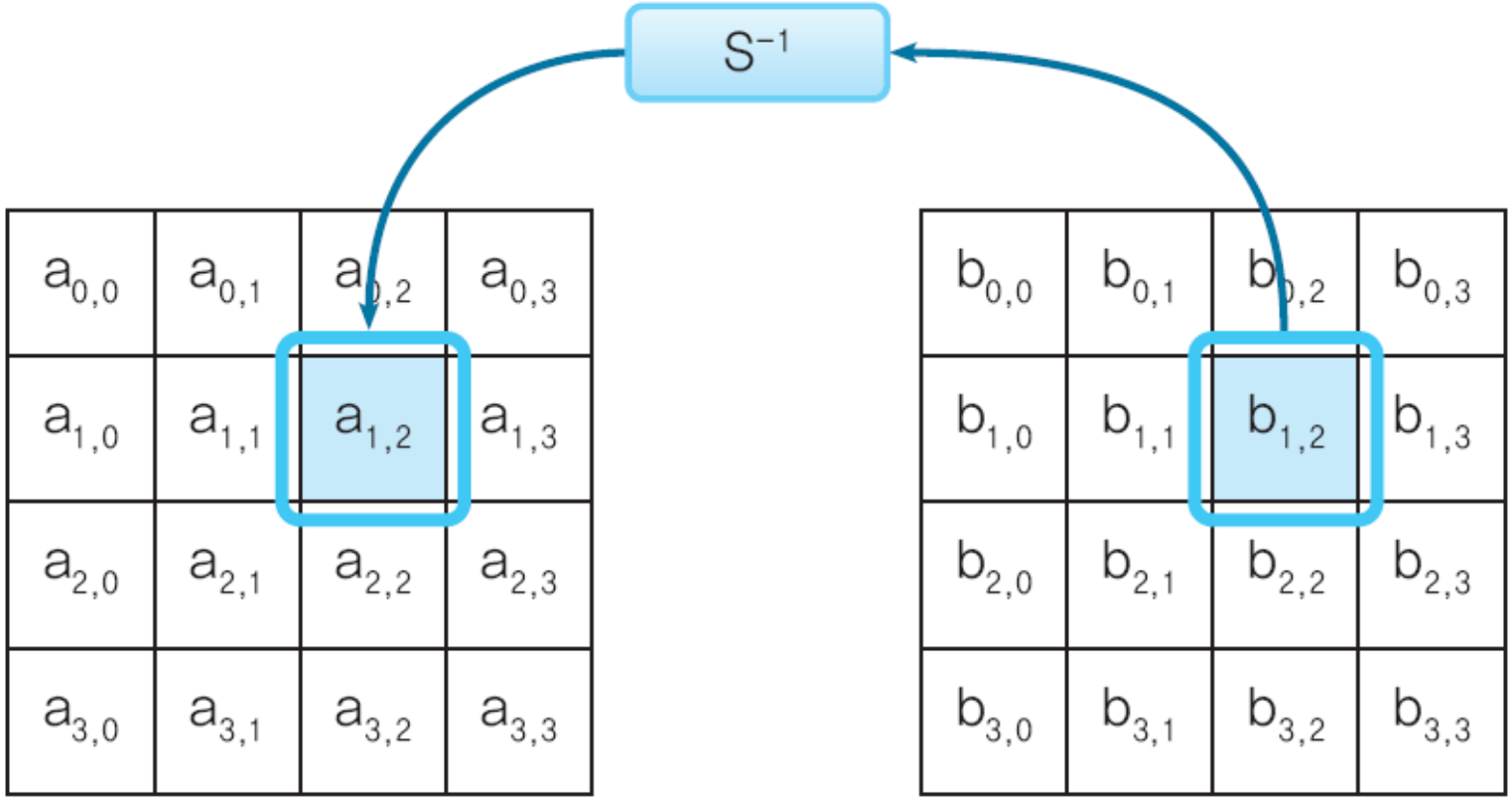


그림 4-17 • InvSubBytes(역 바이트 대체)

6.3 Rijndael 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
- **SEED 및 ARIA**
  - 우리나라 표준
- **AES(Rijndael)**
  - 고속
  - 다양한 플랫폼
  - 현재 까지 안전
  - 사용 권장
  - AES 최종 후보 5개도 사용가능

Quiz 3 대칭암호에서 키의 비트 길이는 얼마만큼 필요한가?

지금 당신이 이용할 수 있는 컴퓨터 파워를 다음과 같이 가정한다.

- 컴퓨터 1대는 1초간에 10^{20}개의 키를 시험할 수 있다.
- 컴퓨터는 10^{100}대 존재한다.
- 전체 컴퓨터는 10^{20}년을 움직인다.

이 정도의 컴퓨터 파워를 사용해도 키 공간에 속하는 키 전부를 전사 공격으로 조사할 수 없도록 하기 위해서는 키의 비트 길이는 몇 비트면 되는가?

Quiz 4 대칭 암호의 기초 지식

다음 문장 중 바른 것에는 ○, 틀린 것에는 X를 표시하시오.

(1) 대칭 암호에서는 암호화 키와 복호화 키는 같다.

(2) 장래에 컴퓨터의 계산력이 충분히 높아지면 일회용 패드의 암호문을 현실적인 시간 내에 해독할 수 있게 된다.

(3) 키의 길이가 56비트일 때 전사 공격으로 바른 키가 발견되기까지 평균 시행 회수는 약 2^{28}회 이다.

(4) AES는 강한 대칭 암호 알고리즘이지만, 상용으로 이용하기 위해서는 NIST에 대해 특허료를 지불하여야 한다.

(5) 현재 DES는 현실적인 시간 내에 해독할 수 있다.

(6) AES로 선정된 암호 알고리즘은 Rijndael이다.