

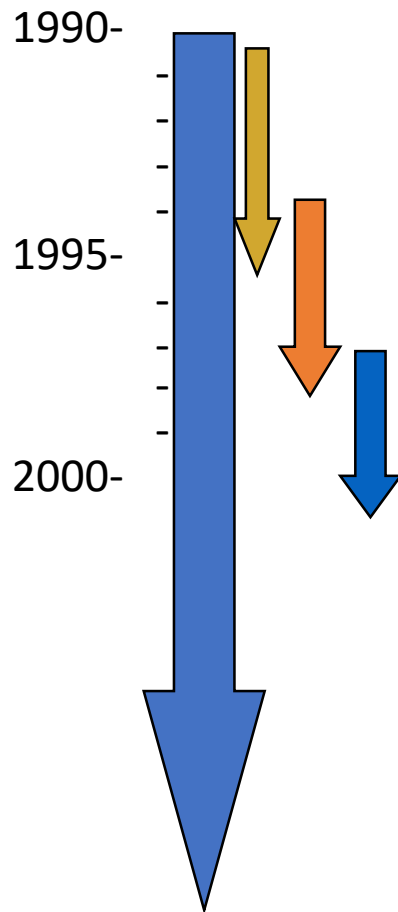
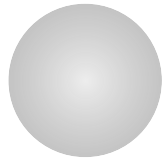
Cryptanalysis (암호분석)

AES – Advanced Encryption Standard
Variants of Differential Cryptanalysis

2020.6

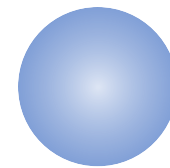
목차

- ▶ Block cipher AES
- ▶ Integral Cryptanalysis
- ▶ Impossible Differential Attack

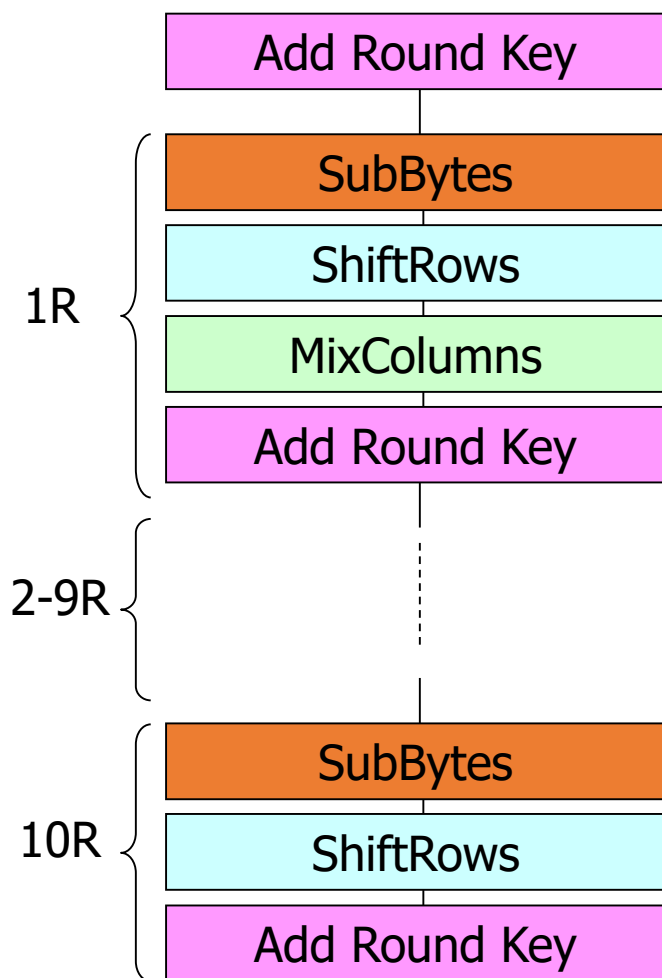


AES

- PhD Daemen
- PhD Rijmen
- AES process
- 1996: Shark
- 1997: Square
- 1998: BKSQ, Rijndael



AES 알고리즘



▶ 알고리즘 구조

- ▶ SPN(Substitution Permutation Network)
- ▶ 10라운드

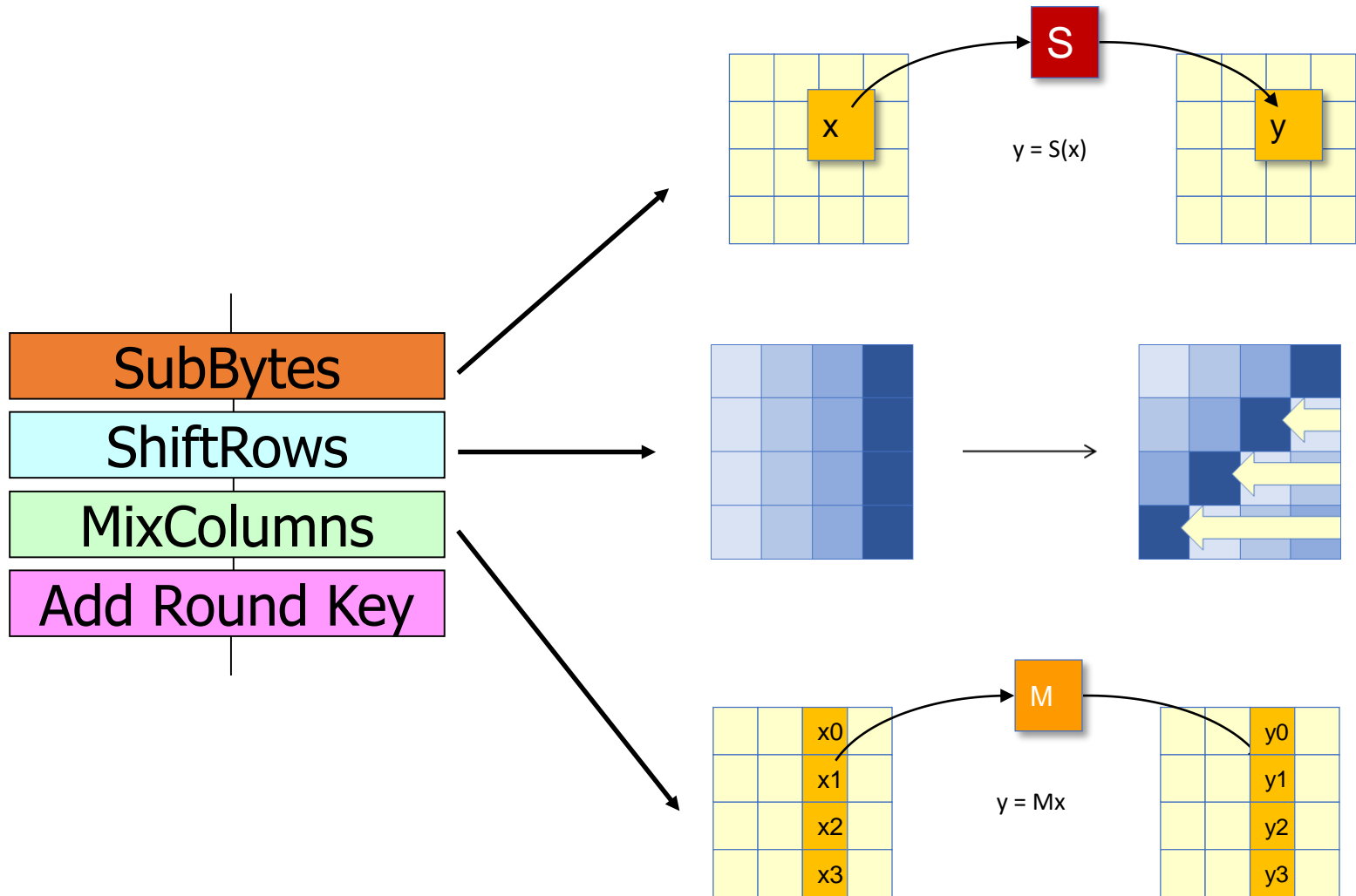
▶ 라운드 구성요소

- ▶ Add Round Key
- ▶ SubBytes
- ▶ ShiftRows
- ▶ MixColumns

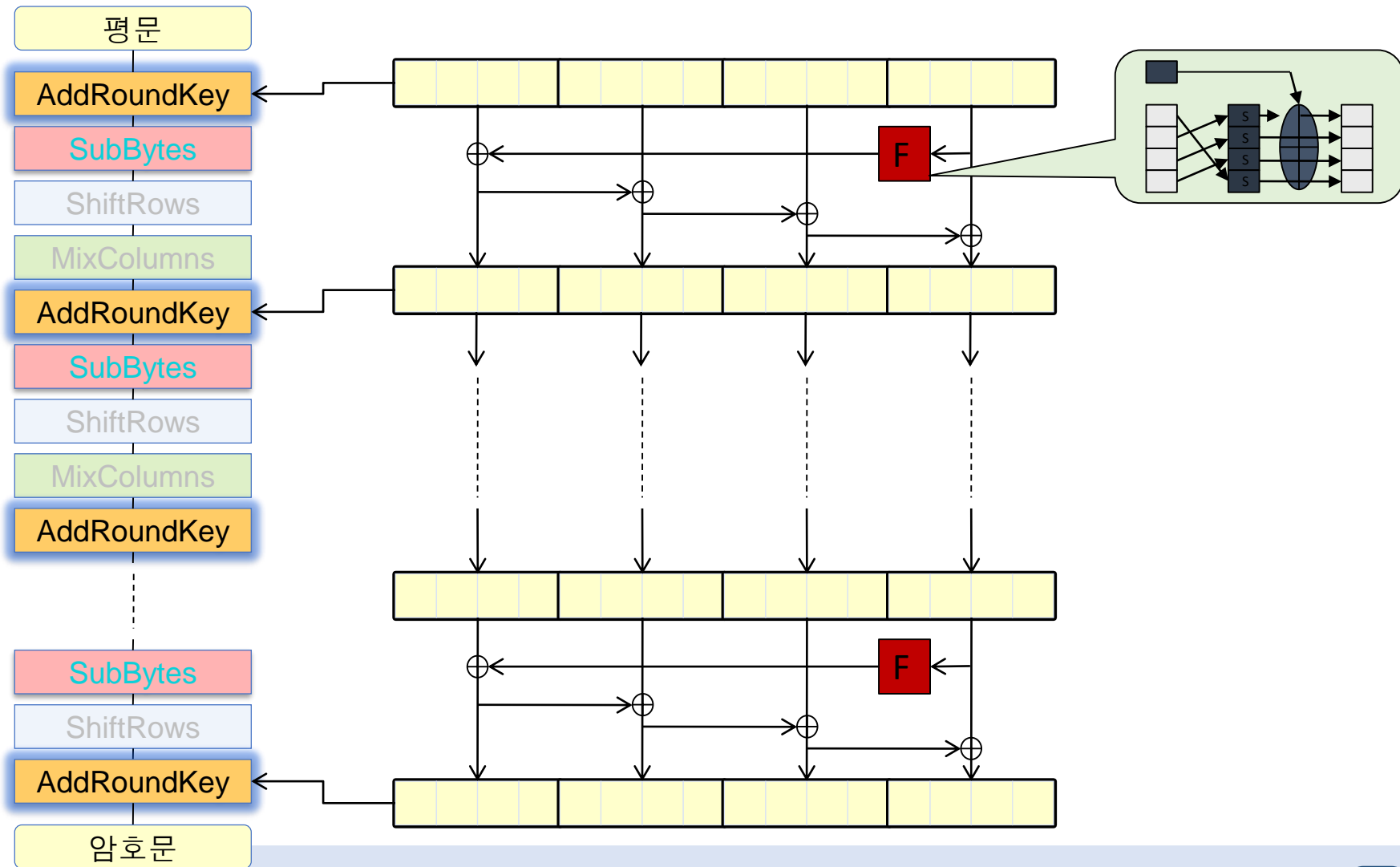
▶ 마지막 라운드

- ▶ MixColumns를 수행하지 않음
- ▶ 구현 효율성과 암호/복호 대칭성에 유리하기 때문

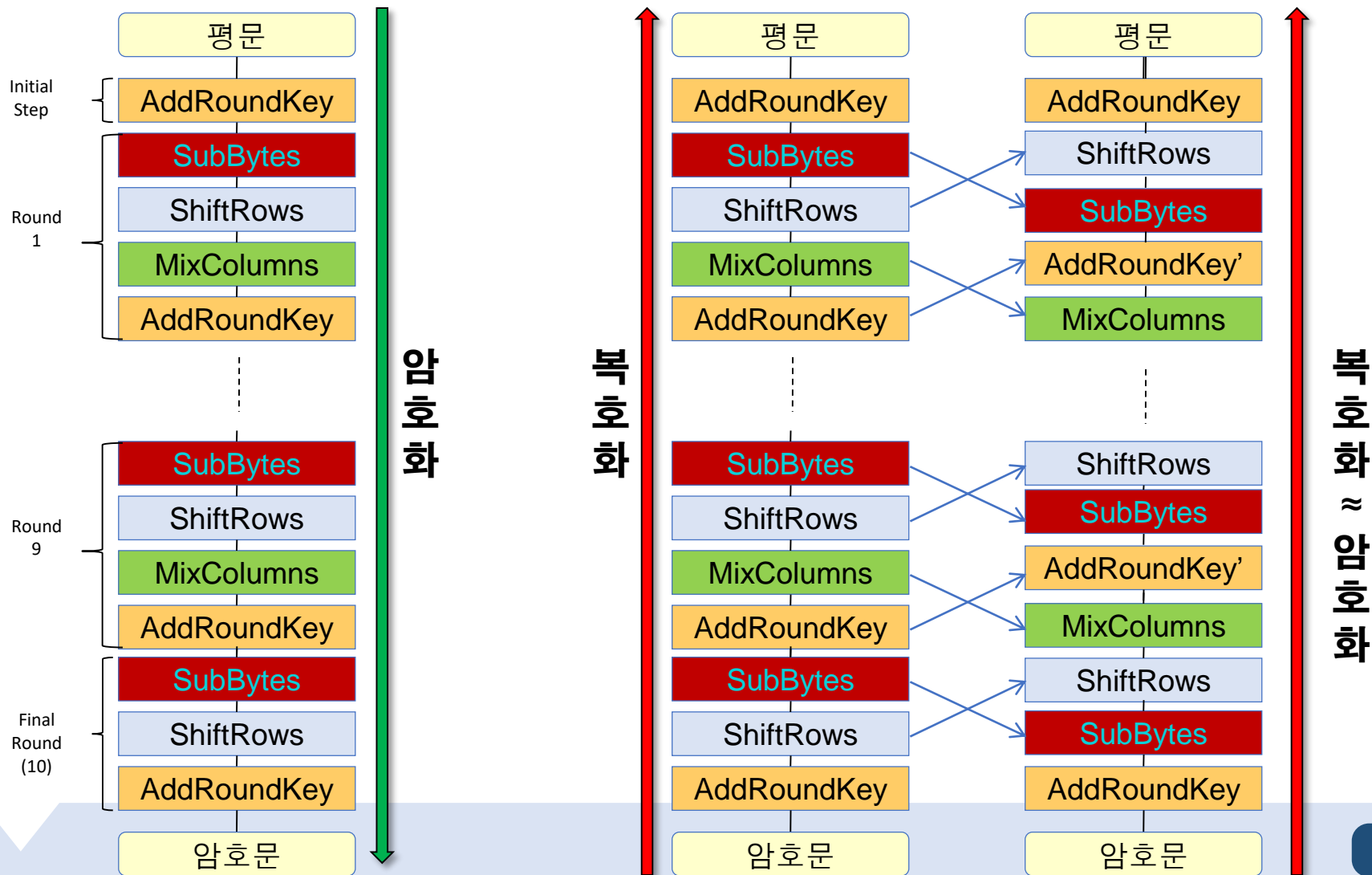
AES 라운드 함수



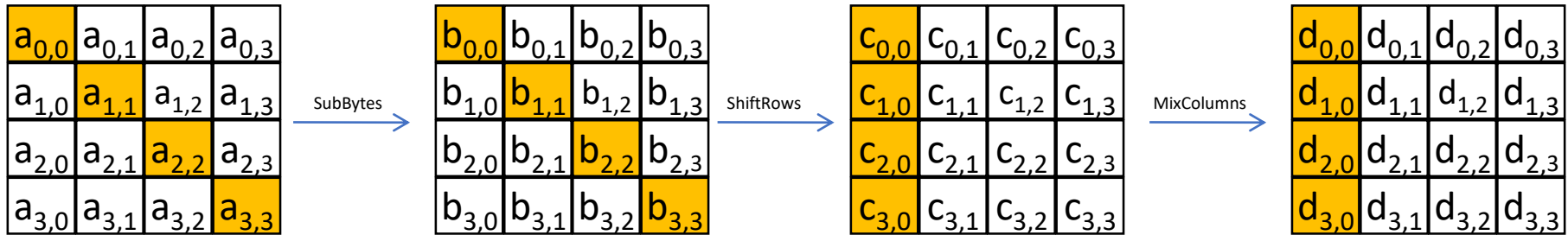
AES의 구조(5) - 키 스케줄



AES의 대칭구조



AES 라운드 함수의 테이블 구현(1)

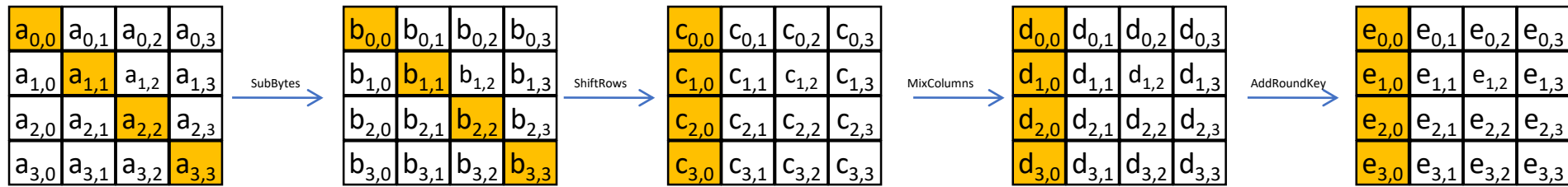


$$\begin{pmatrix} d_{00} \\ d_{10} \\ d_{20} \\ d_{30} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} S(a_{00}) \\ S(a_{11}) \\ S(a_{22}) \\ S(a_{33}) \end{pmatrix}$$

$$= S(a_{00}) \begin{pmatrix} 02 \\ 01 \\ 01 \\ 03 \end{pmatrix} + S(a_{11}) \begin{pmatrix} 03 \\ 02 \\ 01 \\ 01 \end{pmatrix} + S(a_{22}) \begin{pmatrix} 01 \\ 03 \\ 02 \\ 01 \end{pmatrix} + S(a_{33}) \begin{pmatrix} 01 \\ 01 \\ 03 \\ 02 \end{pmatrix}$$

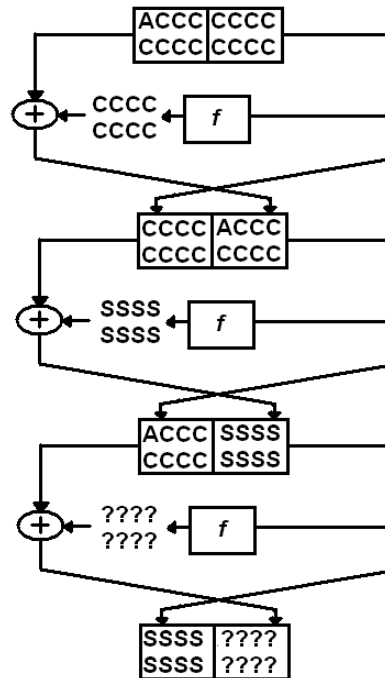
$$= \mathbf{T_0}(a_{00}) + \mathbf{T_1}(a_{11}) + \mathbf{T_2}(a_{22}) + \mathbf{T_3}(a_{33})$$

AES 라운드 함수의 테이블 구현(2)



$$\begin{pmatrix} e_{00} \\ e_{10} \\ e_{20} \\ e_{30} \end{pmatrix} = T_0(a_{00}) + T_1(a_{11}) + T_2(a_{22}) + T_3(a_{33}) + \begin{pmatrix} rk_{00} \\ rk_{10} \\ rk_{20} \\ rk_{30} \end{pmatrix}$$

$$T_0(x) = S(x) \begin{pmatrix} 02 \\ 01 \\ 01 \\ 03 \end{pmatrix}, \quad T_1(x) = S(x) \begin{pmatrix} 03 \\ 02 \\ 01 \\ 01 \end{pmatrix}, \quad T_2(x) = S(x) \begin{pmatrix} 01 \\ 03 \\ 02 \\ 01 \end{pmatrix}, \quad T_3(x) = S(x) \begin{pmatrix} 01 \\ 01 \\ 03 \\ 02 \end{pmatrix}$$



Integral Cryptanalysis

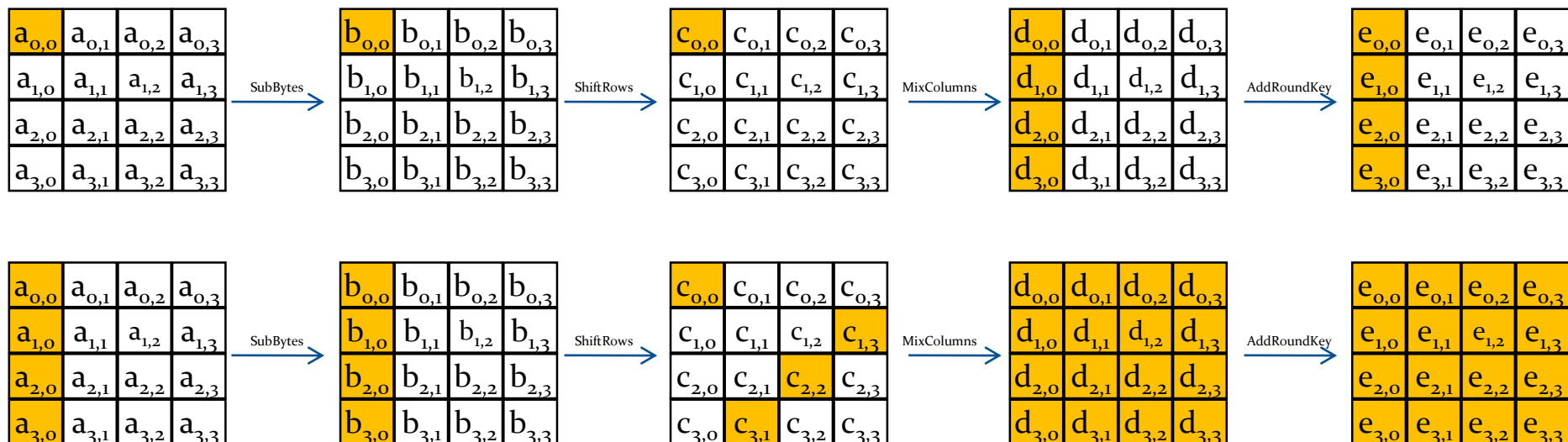
Integral cryptanalysis 개요

- ▶ Square attack
 - ▶ 블록암호 Square 전용 공격기법
 - ▶ Rijndael 분석에 사용(설계자의 안전성 분석)
 - ▶ 특징: 확률 1의 distinguisher를 사용하는 공격
- ▶ Integral cryptanalysis
 - ▶ 2002년 Knudsen에 의하여 제안
 - ▶ Square attack, Saturation attack 등의 다양한 이름을 통합한 결과
- ▶ 바이트 단위 연산 중심의 블록암호 공격에 효과적
 - ▶ 적용 암호: Square, AES, Camellia, Safer++, ARIA 등

Basic Idea (1/2)

▶ AES의 차분전파 특성

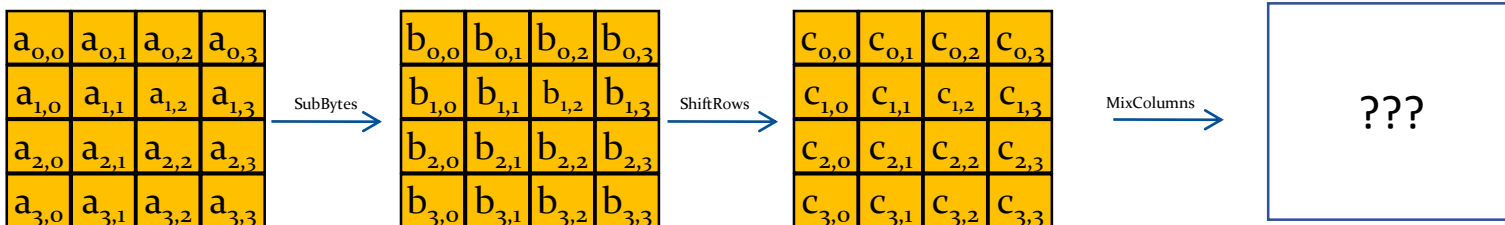
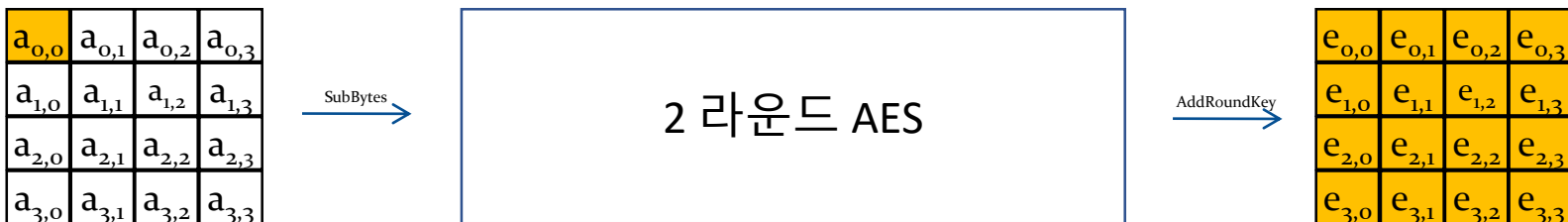
- ▶ 첫 바이트($a_{0,0}$)만 0이 아닌 차분이 있는 경우
- ▶ 1 라운드 후, 4바이트만 영향 (나머지는 차분=0)
- ▶ 2 라운드 후, 16바이트 모두에 영향



Basic Idea (2/2)

▶ AES의 차분전파 특성

▶ 3 라운드 이후는 어떻게 될까?



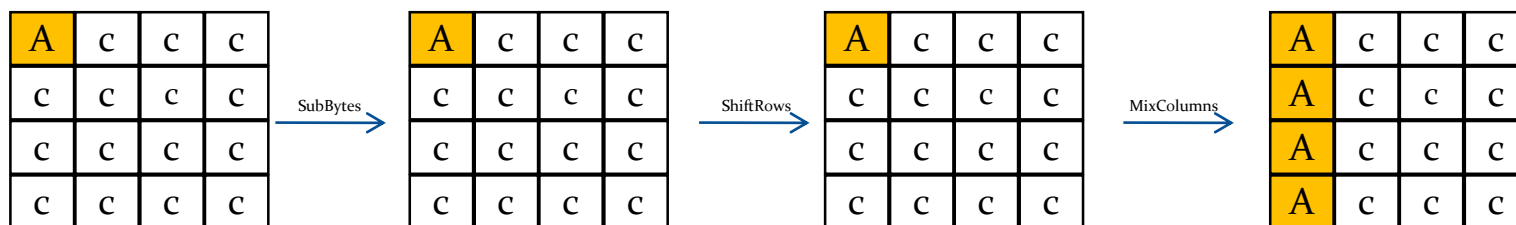
▶ 3 라운드 MixColumns 전까지 모든 바이트 차분이
nonzero

Lambda-set: state vector

- ▶ State vector
 - ▶ 평문, 암호문, 중간단계 블록
- ▶ Λ -set: 256개의 state vector
 - ▶ (특정) 한 바이트: 256가지의 모든 값 (**Active**)
 - ▶ 나머지 15바이트는 고정된 값 (**Fixed**)
- ▶ Λ -set의 예: v_0 (Active), v_1, \dots, v_{15} (Fixed)
 $\{(v_0, v_1, \dots, v_{15}) \mid v_0 = 0, 1, 2, \dots, 255, v_1 = c_1, \dots, v_{15} = c_{15}\}$

Operations on Lambda-set

▶ Λ -set의 라운드 함수 적용



▶ 바이트 단위의 일대일 함수는
Active, const 성질을 보존한다.

$$\begin{bmatrix} d_{0,0} \\ d_{1,0} \\ d_{2,0} \\ d_{3,0} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} c_{0,0} \\ c_{1,0} \\ c_{2,0} \\ c_{3,0} \end{bmatrix}$$

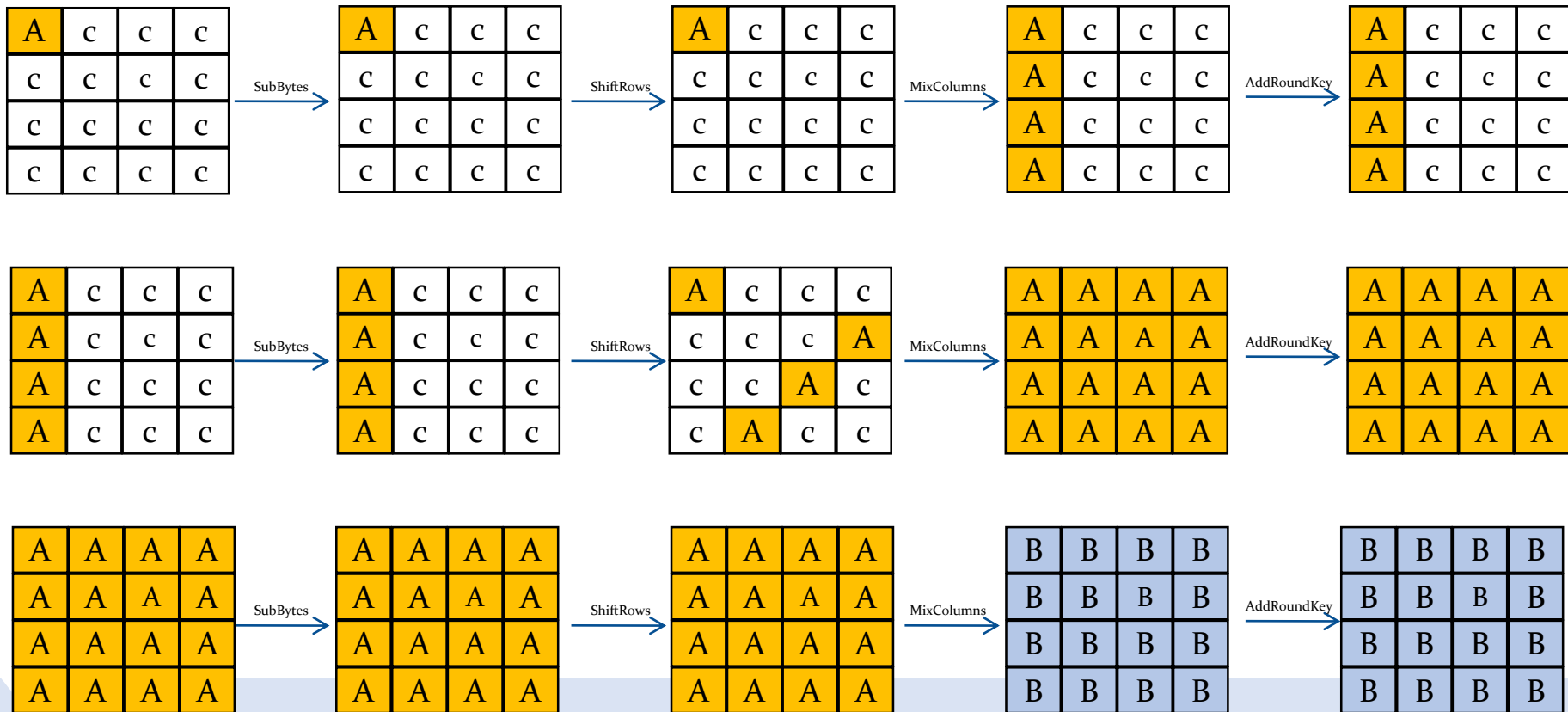
XOR on Lambda-set

- ▶ Λ -set의 각 바이트(Active, Fixed) 표기법
 - ▶ A (Active): 256가지 모든 값을 가지는 바이트
 - ▶ c (const, Fixed): 고정된 하나의 값만 가지는 바이트
 - ▶ **B (Balanced)**: 모든 값(256개, 중복가능)을 XOR하면 0이 되는 바이트
- ▶ Λ -set의 성질
 - ▶ A(Active), c(const) 바이트는 각각 Balanced 바이트

XOR	Active	const	Balanced
Active	Balanced	Active	Balanced
const	Active	Const	Balanced
Balanced	Balanced	Balanced	Balanced

Integral distinguisher

▶ 3 round distinguisher (확률 1)



Basic attack on 4R

▶ 4라운드 integral attack

A	c	c	c
c	c	c	c
c	c	c	c
c	c	c	c

SubBytes

3 라운드 AES

AddRoundKey

B	B	B	B
B	B	B	B
B	B	B	B
B	B	B	B

B	B	B	B
B	B	B	B
B	B	B	B
B	B	B	B

SubBytes

?	?	?	?
?	?	?	?
?	?	?	?
?	?	?	?

ShiftRows

?	?	?	?
?	?	?	?
?	?	?	?
?	?	?	?

AddRoundKey

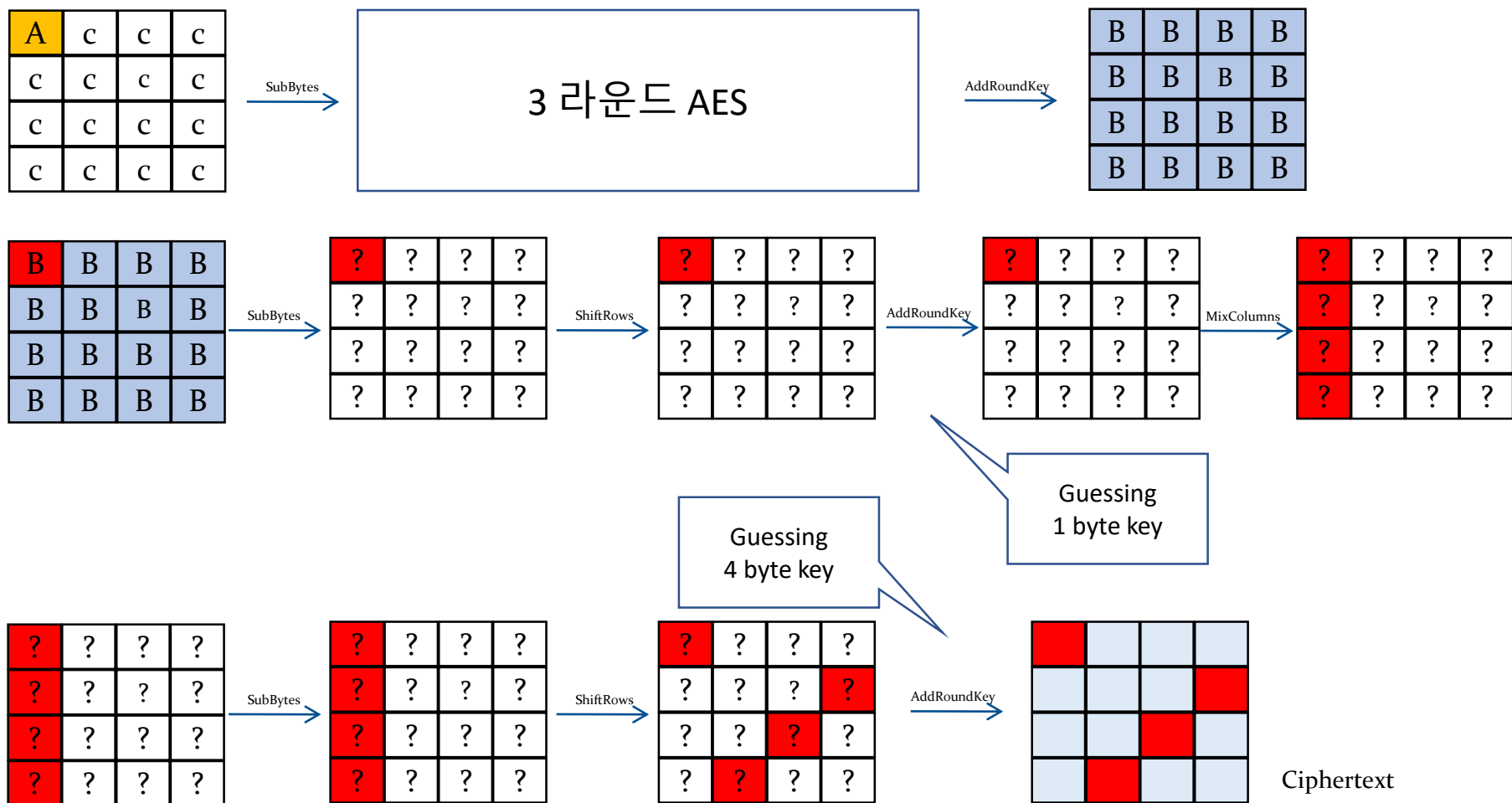
Ciphertext

Check whether it is
balanced or not

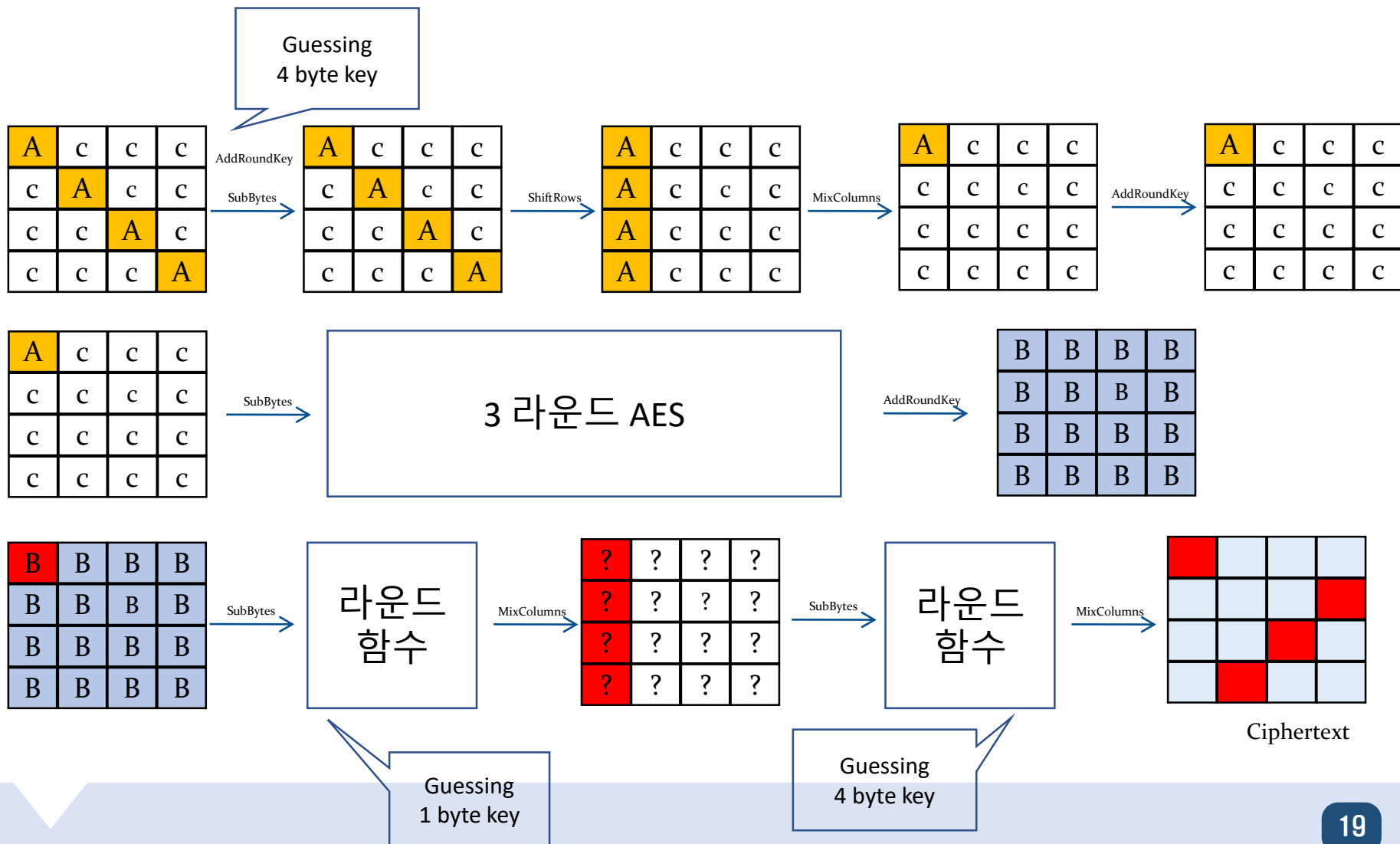
Guessing
1 byte key

RoundKey

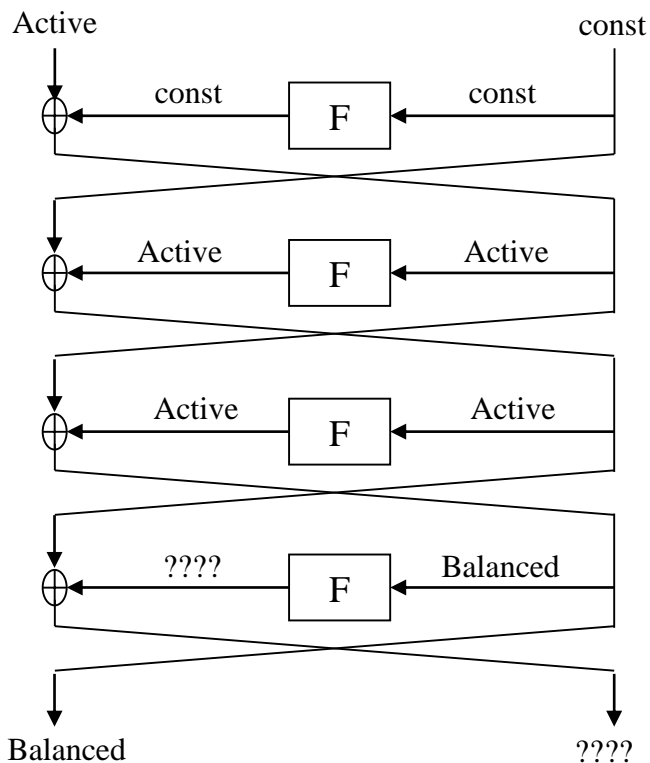
5 Round attack



6 Round attack



Feistel 구조의 Integral distinguisher



▶ 대상

- ▶ 1-1 라운드함수의 Feistel 구조
- ▶ 라운드함수의 입력비트: m

▶ 4R distinguisher

- ▶ 입력평문: (Active, const)
- ▶ Active: 2^m 개의 서로 다른 값으로 구성
- ▶ 4라운드 출력은 (Balanced, ????)

요약

▶ Integral cryptanalysis

- ▶ 확률 1의 distinguisher를 구성
- ▶ Distinguisher의 전후 라운드 키를 예측하는 방법으로 공격

▶ AES-128의 분석 결과

- ▶ 6라운드까지는 전수조사보다 좋은 공격이 가능함
- ▶ 7라운드 공격도 가능하나 거의 전수조사와 같음
- ▶ AES의 발표(제안) 당시 가장 효과적인 분석기법

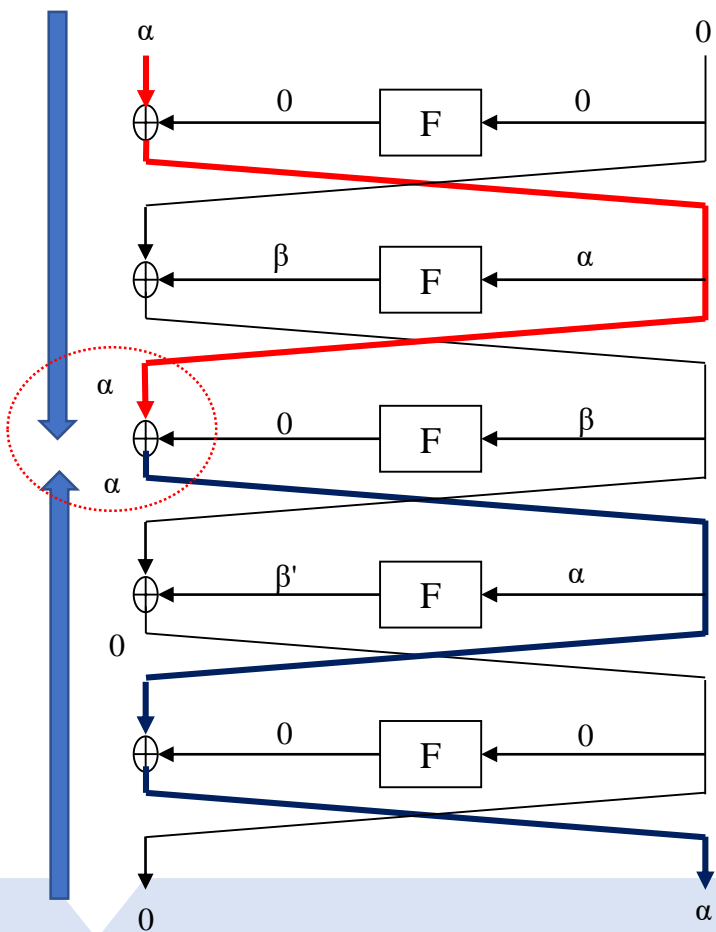


Impossible Differential Attack

Impossible differential cryptanalysis 개요

- ▶ 불능 차분(Impossible differential)
 - ▶ 발생할 확률이 0인 입출력 차분 특성
 - ▶ 불능 차분을 유도하는 암호키를 지워가는 방식으로 공격
 - ▶ 특징: 확률 0의 distinguisher를 사용하는 공격
- ▶ AES의 불능 차분 공격
 - ▶ Biham, Keller (1999): 5라운드
 - ▶ Cheon, et al. (2002): 6라운드
 - ▶ JS. Kim, et al. (2008): 7라운드
- ▶ 바이트 단위 연산 중심의 블록암호 공격에 효과적
 - ▶ Square, AES, Camellia, ARIA 등

Feistel 구조의 Impossible Differential distinguisher



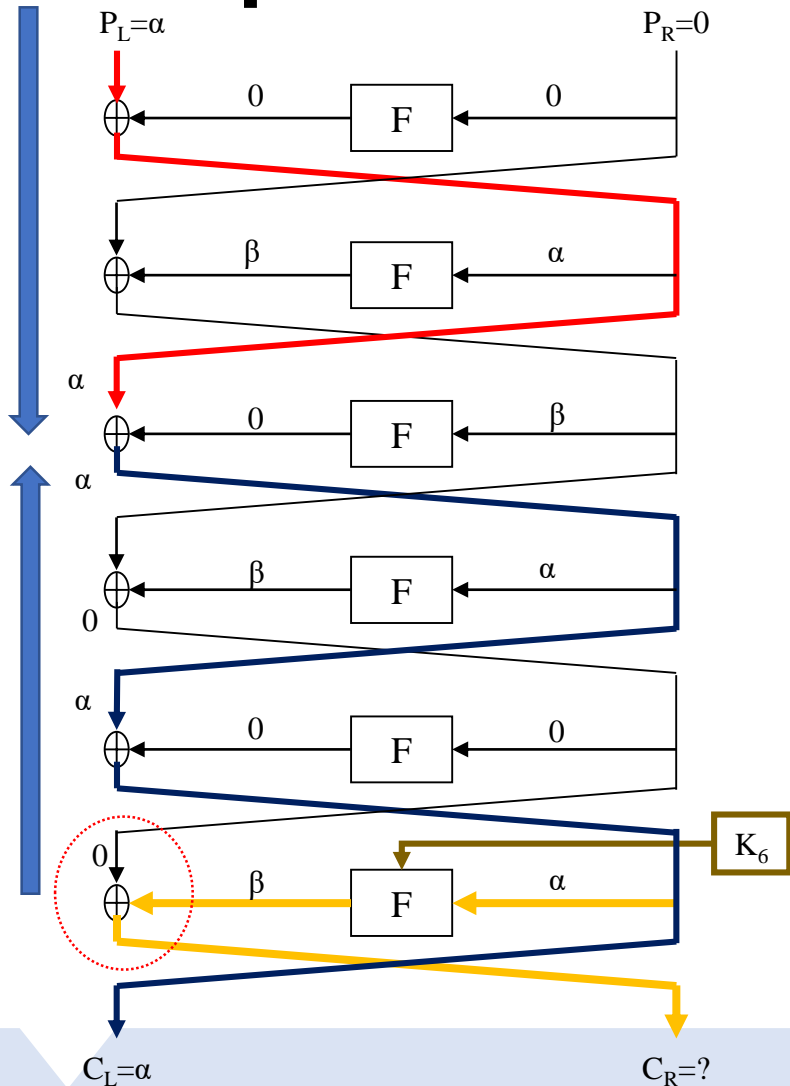
▶ 대상

- ▶ 1-1 라운드함수의 Feistel 구조

▶ 5R distinguisher

- ▶ 입력 차분: $(\alpha, 0)$
- ▶ 5라운드 출력 차분이 $(0, \alpha)$ 가 될 수 없다.
- ▶ 모순: 3라운드 출력차분
 - ▶ 위→아래: 0이 아닌 차분
 - ▶ 아래→위: 차분 0

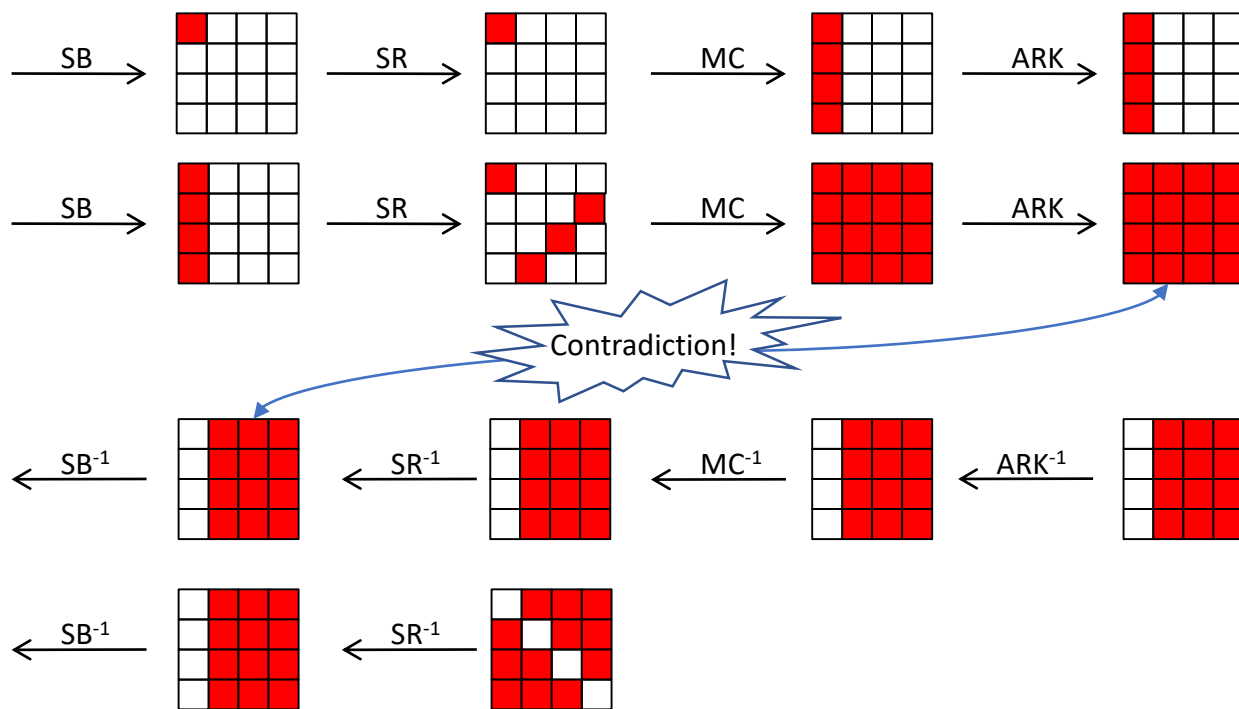
6R Feistel 암호의 Impossible Differential Attack



- ▶ 평문 structure (블록크기 $2n$ 비트)
 - ▶ 2^n 개 평문: $(P_L, P_R) = (\text{Active}, \text{const})$
 - ▶ 가능한 평문 쌍: 약 2^{2n-1}
- ▶ 평문-암호문 쌍의 필터링
 - ▶ 평문-암호문 쌍 $(P_L, P_R) : (C_L, C_R)$ 가
운데 다음 조건을 만족하는 것만 수집
 $P_L^{(1)} \oplus P_L^{(2)} = C_L^{(1)} \oplus C_L^{(2)}$
 (structure 구성방법에서 $P_R^{(1)} \oplus P_R^{(2)} = 0$)
 $2^{2n-1} * 2^{-n} = 2^{n-1}$ 개의 쌍이 남음
- ▶ 6라운드 키 K_6 (k비트) 예측
 - ▶ 예측한 K_6 와 (C_L, C_R) 로부터 5라운드 입력차
분을 계산
 - ▶ 5라운드 왼쪽이 0이면, Wrong Key로 판단
 - ▶ 필터링 된 모든 쌍에 대하여 Wrong key로
걸러지지 않으면 키 후보로 등록
 (한 structure 당 $2^k * (1 - 2^{-n})^{(2^{n-1})} =$
 $2^k * e^{(-1/2)}$ 개의 Wrong key가 발견됨)

4R AES의 impossible differential distinguisher

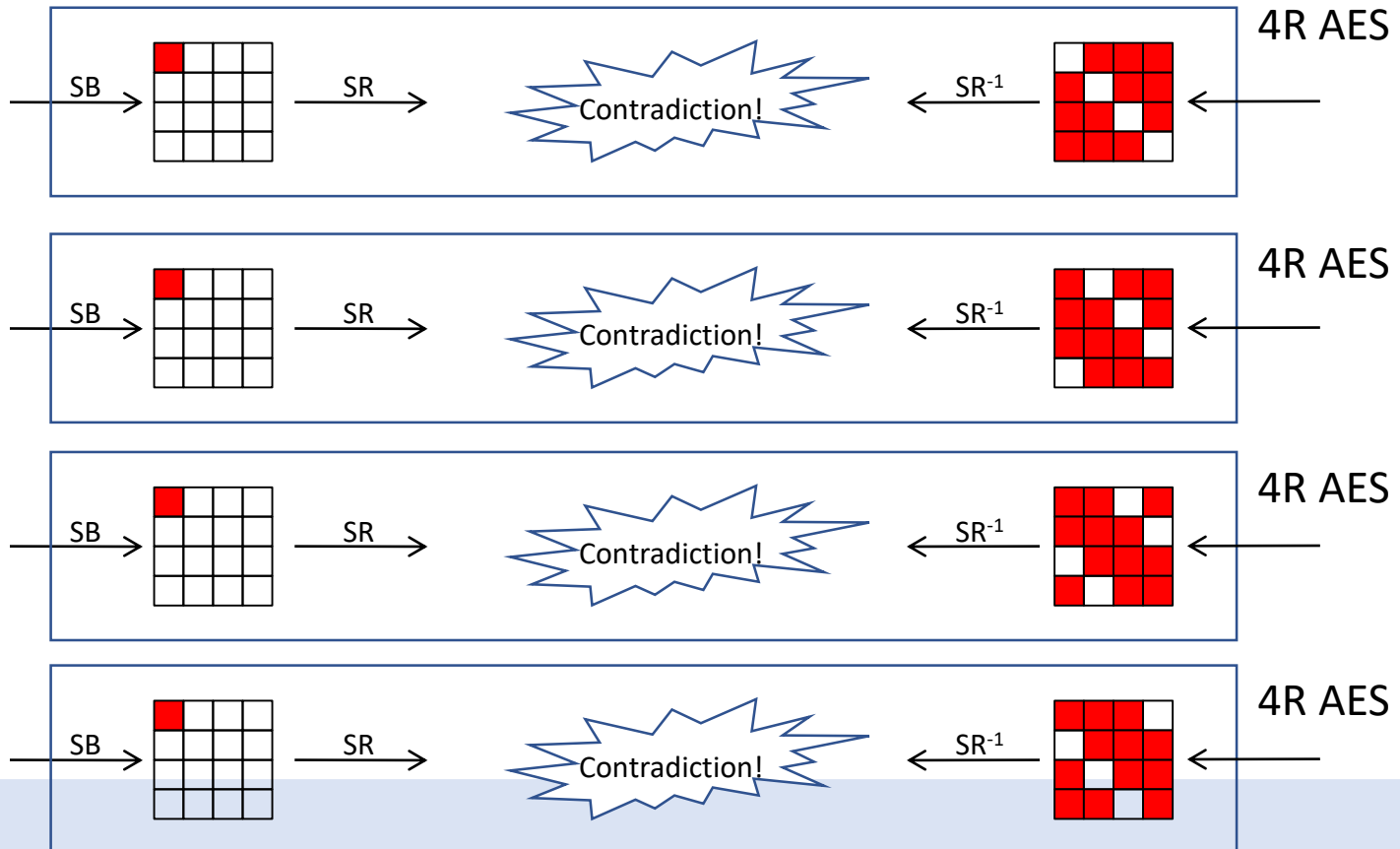
▶ 4R AES의 불능 차분 특성



4R AES의 4가지 distinguishers

▶ 불능 차분 distinguisher

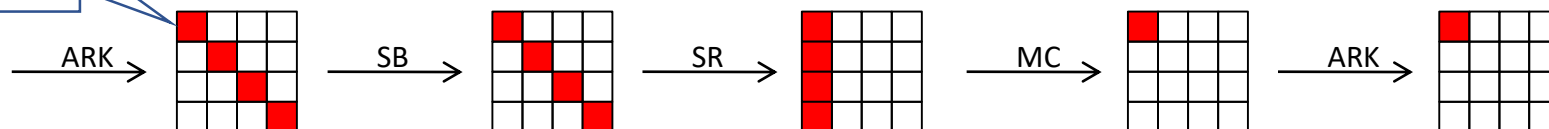
- ▶ 랜덤함수인 경우 만족할 확률 = 2^{-30}
- ▶ AES의 4R인 경우 만족할 확률 = 0



Basic Attack (IDC)

- ▶ $5R = 1R + 4R$ (distinguisher)에 대한 공격 방법
 - ▶ 선택평문 2^{32} 개의 Structure 구성: (0,5,10,15)외는 고정값
 - ▶ 1R 4바이트 키 예측 → 4R ID를 따르는 경우 키 후보에서 배제

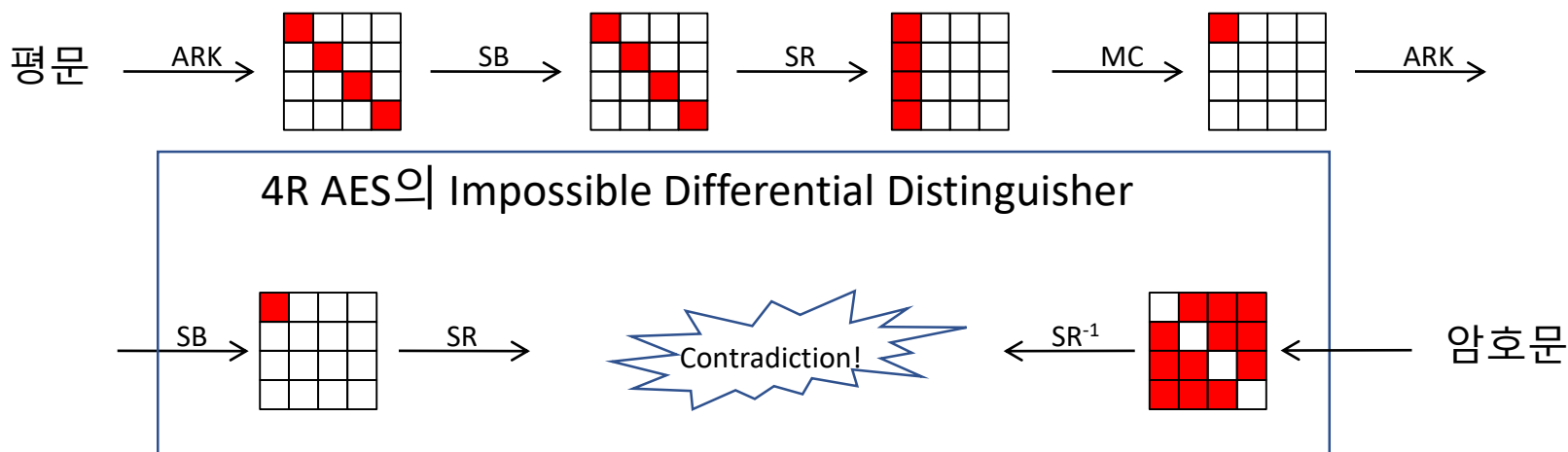
Guessing
4 byte key



4R AES의 Impossible Differential Distinguisher



Basic Attack (IDC) Step by Step



▶ 공격 알고리즘

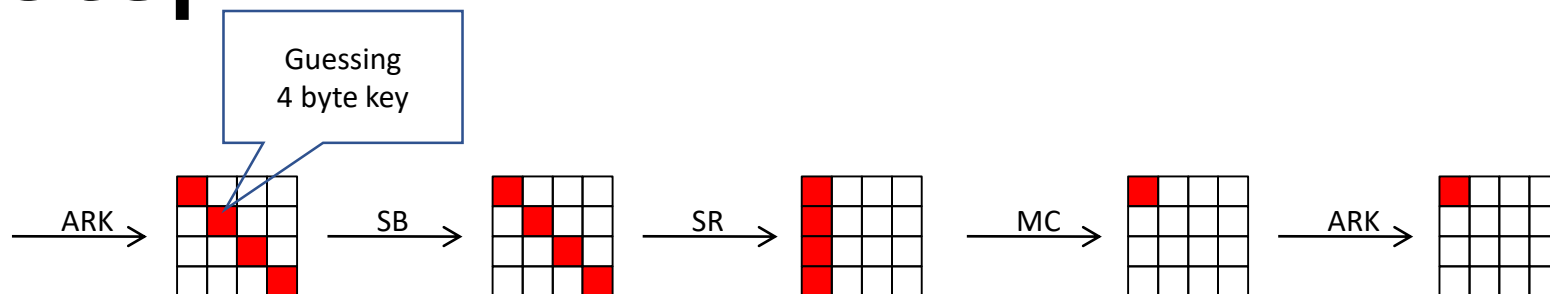
- ▶ 선택평문 2^{32} 개의 Structure: (0,5,10,15)외는 고정값

→ 2^{63} 평문-암호문 쌍

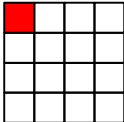
- ▶ 필터링

→ 5R 암호화 결과가  인 평문-암호문 쌍 → $2^{63}/2^{32} = 2^{31}$ 쌍

Basic Attack (IDC) Step by Step



▶ 공격 알고리즘

- ▶ 1라운드 암호키 4바이트 예측 (남은 평문-암호문 2^{31} 쌍에 대하여 적용)
- ▶ 각 평문쌍에 예측된 암호키를 적용하여 1라운드 출력차분을 계산
→ 차분의 패턴이  이면, wrong key로 판정 (한번이라도 발생하면)
- ▶ Wrong Key가 후보로 남을 확률 (한번도 2라운드 입력차분이 ID콜 아님)
→ $(1 - 2^{-24})^{(2^{31})} = [\{ (1 - 2^{-24})^{(-2^{24})} \}^{(-1)}]^{(2^7)} = e^{(-2^7)}$
- ▶ Right Key: 항상 후보로 남음(위의 차분이 나올 수 없음)