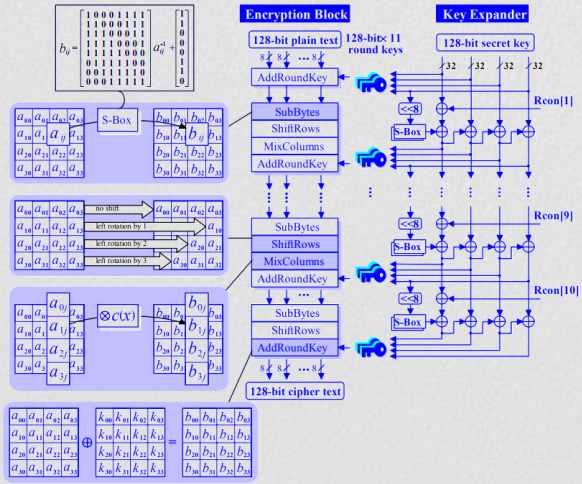
04

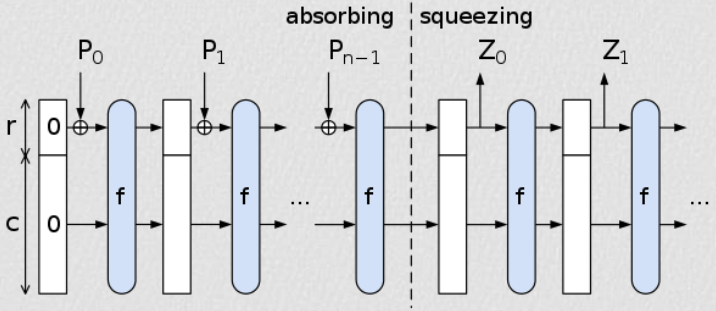
# PQC 관련 암호 최적화

# 양자 내성 암호 (공개키 암호) 관련 암호

## 양자 내성 암호의 활용과 관련 암호

- Key Encapsulation Mechanism (KEM), Digital Signature
  - 난수생성기: AES, SHAKE (SHA-3)
  - 메시지 다이제스트: Hash (SHA-2, SHA-3)

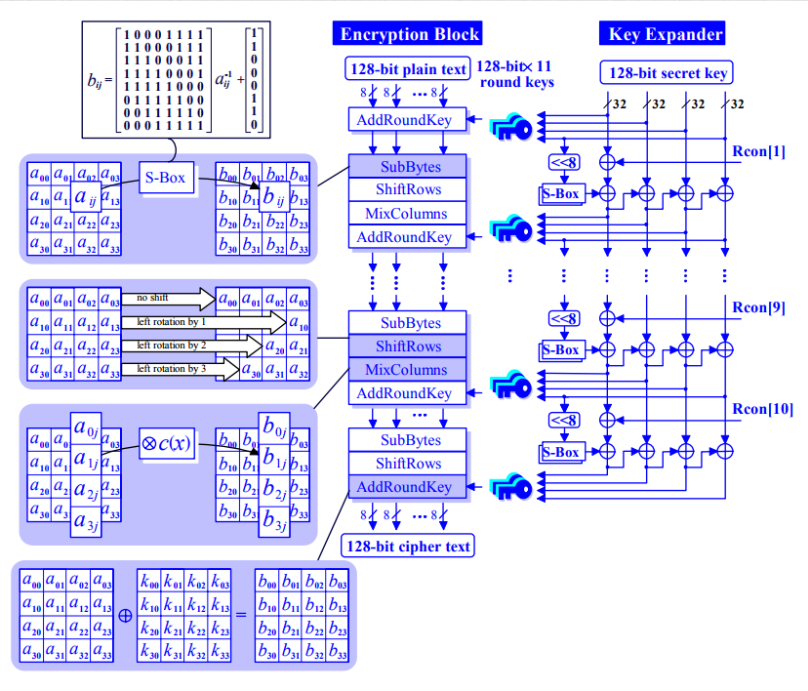


# AES 암호화

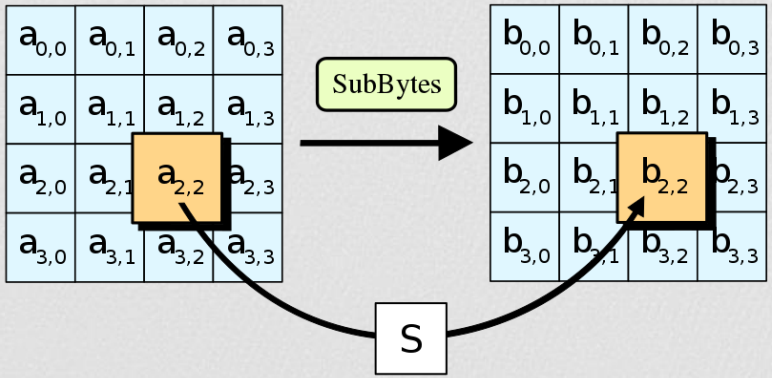
## 국제 블록암호 표준: AES

- 1998년도 제안된 블록 암호화 (128/192/256-비트 보안 제공)
- SPN 구조를 가진 대표적인 블록 암호



# AES 암호화 - SubBytes (LUT 기반)

## 8-비트 입력값을 8-비트 출력값으로 치환




| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0a | 0b | 0c | 0d | 0e | 0f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | 63 | 7c | 77 | 7b | f2 | 6b | 6f | c5 | 30 | 01 | 67 | 2b | fe | d7 | ab | 76 |
| 10 | ca | 82 | c9 | 7d | fa | 59 | 47 | f0 | ad | d4 | a2 | af | 9c | a4 | 72 | c0 |
| 20 | b7 | fd | 93 | 26 | 36 | 3f | f7 | cc | 34 | a5 | e5 | f1 | 71 | d8 | 31 | 15 |
| 30 | 04 | c7 | 23 | c3 | 18 | 96 | 05 | 9a | 07 | 12 | 80 | e2 | eb | 27 | b2 | 75 |
| 40 | 09 | 83 | 2c | 1a | 1b | 6e | 5a | a0 | 52 | 3b | d6 | b3 | 29 | e3 | 2f | 84 |
| 50 | 53 | d1 | 00 | ed | 20 | fc | b1 | 5b | 6a | cb | be | 39 | 4a | 4c | 58 | cf |
| 60 | d0 | ef | aa | fb | 43 | 4d | 33 | 85 | 45 | f9 | 02 | 7f | 50 | 3c | 9f | a8 |
| 70 | 51 | a3 | 40 | 8f | 92 | 9d | 38 | f5 | bc | b6 | da | 21 | 10 | ff | f3 | d2 |
| 80 | cd | 0c | 13 | ec | 5f | 97 | 44 | 17 | c4 | a7 | 7e | 3d | 64 | 5d | 19 | 73 |
| 90 | 60 | 81 | 4f | dc | 22 | 2a | 90 | 88 | 46 | ee | b8 | 14 | de | 5e | 0b | db |
| a0 | e0 | 32 | 3a | 0a | 49 | 06 | 24 | 5c | c2 | d3 | ac | 62 | 91 | 95 | e4 | 79 |
| b0 | e7 | c8 | 37 | 6d | 8d | d5 | 4e | a9 | 6c | 56 | f4 | ea | 65 | 7a | ae | 08 |
| c0 | ba | 78 | 25 | 2e | 1c | a6 | b4 | c6 | e8 | dd | 74 | 1f | 4b | bd | 8b | 8a |
| d0 | 70 | 3e | b5 | 66 | 48 | 03 | f6 | 0e | 61 | 35 | 57 | b9 | 86 | c1 | 1d | 9e |
| e0 | e1 | f8 | 98 | 11 | 69 | d9 | 8e | 94 | 9b | 1e | 87 | e9 | ce | 55 | 28 | df |
| f0 | 8c | a1 | 89 | 0d | bf | e6 | 42 | 68 | 41 | 99 | 2d | 0f | b0 | 54 | bb | 16 |

# AES 암호화 - SubBytes (연산 기반)

## S-BOX 연산은 갈로아 필드와 아핀 연산으로 구성

1) $GF(2^8) = GF(2)[x]/(x^8 + x^4 + x^3 + x + 1)$ 상에서의 역치 연산 이후

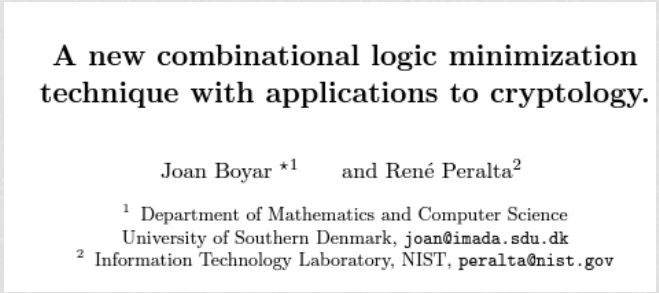
2)
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Affine transformation 수행

# AES 암호화 – SubBytes (연산 기반 최적화)

- **Fermat's theorem**
  - $a^{m-1} = 1 \ (mod\ m)$
  - $a^{m-2} = a^{-1} \ (mod\ m)$ → 곱셈과 제곱 연산을 통해 역치 연산 도출
  - Case study for AES: $a^{-1} = a^{254} = ((a \cdot a^2) \cdot (a \cdot a^2)^4 \cdot (a \cdot a^2)^{16} \cdot a^{64})^2$

- **Bitslicing 기반 구현**
  - Boolean 연산으로 S-box 구현



A new combinational logic minimization technique with applications to cryptology.

Joan Boyar *[1] and René Peralta[2]

[1] Department of Mathematics and Computer Science
University of Southern Denmark, joan@imada.sdu.dk
[2] Information Technology Laboratory, NIST, peralta@nist.gov

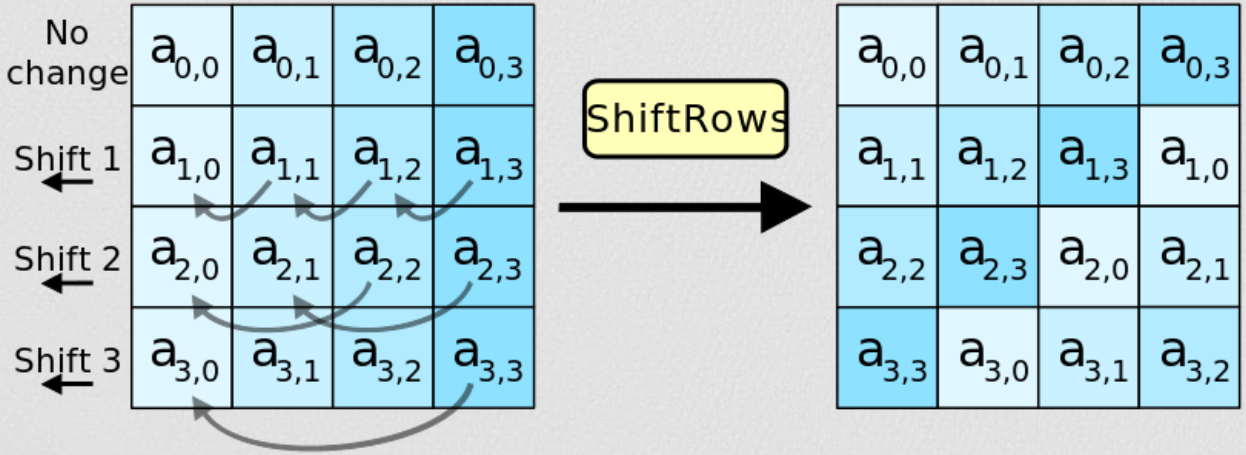
```
T23 = U[7] ^ U[4];
T22 = ~(U[6] ^ U[4]);
T2 = ~(U[7] ^ U[6]);
T1 = U[4] ^ U[3];
T24 = ~(U[3] ^ U[0]);
R5 = U[1] ^ U[0];
T8 = ~(U[6] ^ T23);
T19 = T22 ^ R5;
T9 = ~(U[0] ^ T1);
T10 = T2 ^ T24;
T13 = T2 ^ R5;
T3 = T1 ^ R5;
```

```
T25 = ~(U[5] ^ T1);
R13 = U[6] ^ U[1];
T17 = ~(U[5] ^ T19);
T20 = T24 ^ R13;
T4 = U[3] ^ T8;
R17 = ~(U[5] ^ U[2]);
R18 = ~(U[2] ^ U[1]);
R19 = ~(U[5] ^ U[3]);
Y5 = U[7] ^ R17;
T6 = T22 ^ R17;
T16 = R13 ^ R19;
T27 = T1 ^ R18;
```

```
T15 = T10 ^ T27;
T14 = T10 ^ R18;
T26 = T3 ^ T16;
M1 = T13 & T6;
M2 = T23 & T8;
M3 = T14 ^ M1;
M4 = T19 & Y5;
M5 = M4 ^ M1;
M6 = T3 & T16;
M7 = T22 & T9;
M8 = T26 ^ M6;
```

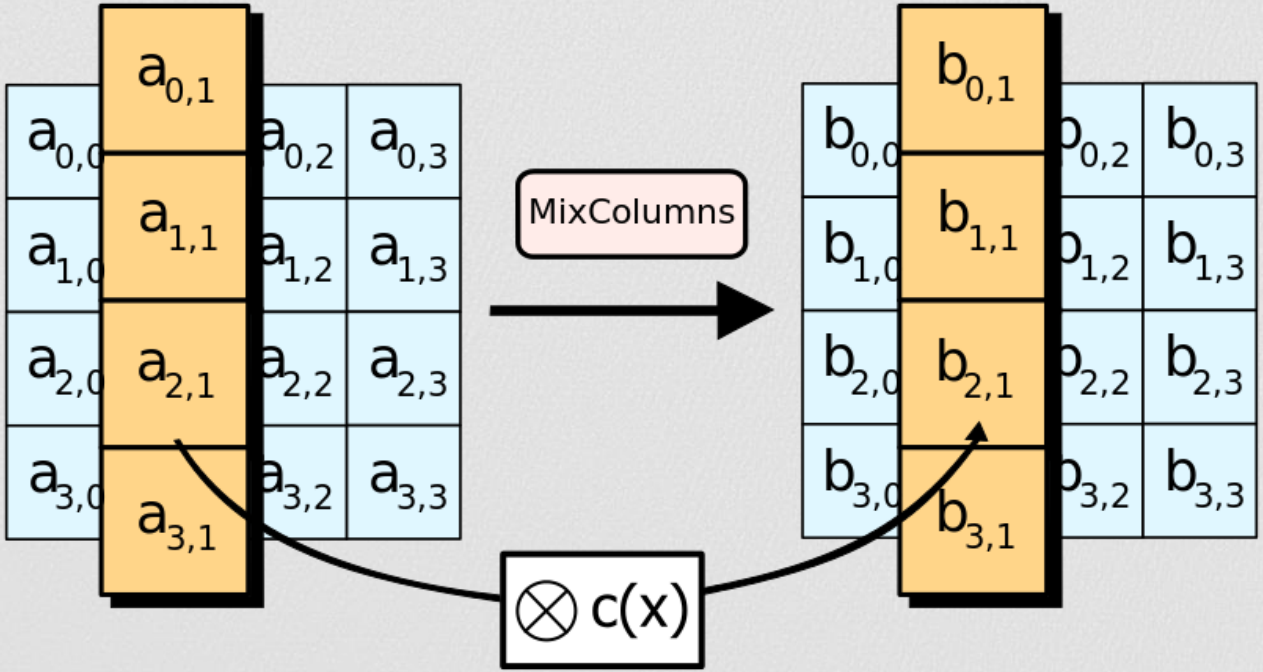
# AES 암호화 - ShiftRows

- AES의 기본 단위는 8-비트 → 예전 컴퓨터는 8-비트가 일반적임
- 하지만 최신 프로세서는 32-비트 이상 → 따라서 8-비트 연산은 비효율적임
  - 이를 해결하는 방법이 T-table 기반 구현
  - bitslicing 기법도 있음 → 다만 바로 적용하는 것은 불가능



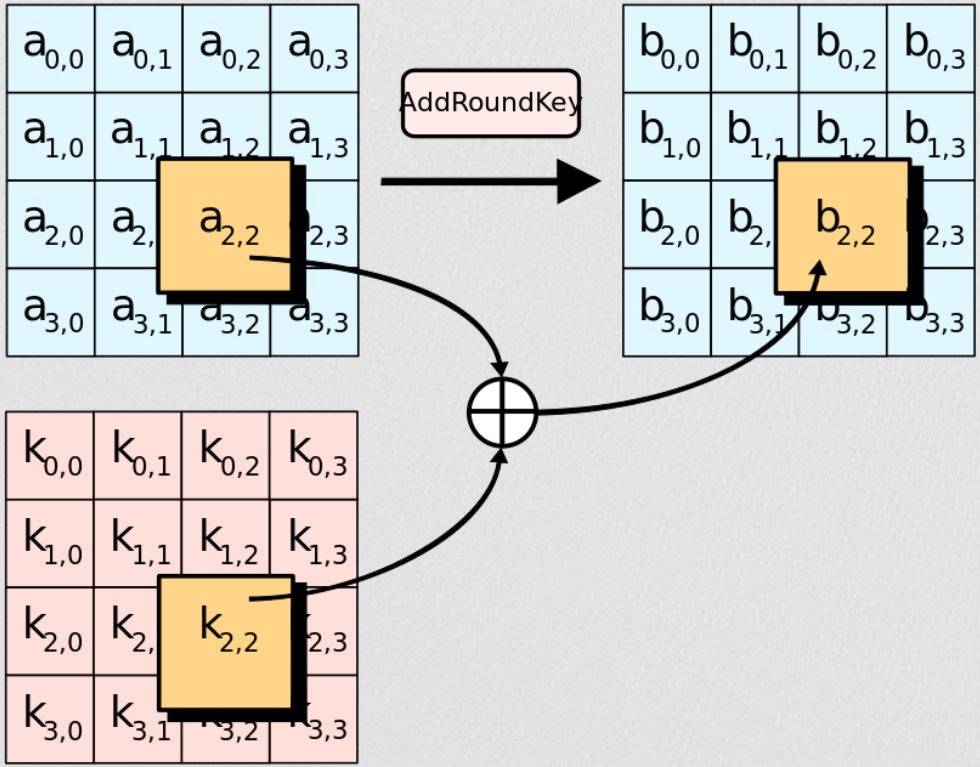
# AES 암호화 - MixColumns

- c값은 1, 2, 혹은 3
- 곱하기 연산이지만 덧셈과 비트시프트로 가능
  (e.g. $a \times 3 = a + a \ll 1$)



# AES 암호화 - AddRoundKey

- 무작위 비밀키를 평문에 더해줌



# AES 암호화 – T-Table 기반 구현

- 32-비트 프로세서 상에서 8-비트 단위 AES를 구현하기 위한 방법론

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}$$

MixColumns

$$T_0[a] = \begin{pmatrix} 2 \cdot S[a] \\ S[a] \\ S[a] \\ 3 \cdot S[a] \end{pmatrix}, T_1[a] = \begin{pmatrix} 3 \cdot S[a] \\ 2 \cdot S[a] \\ S[a] \\ S[a] \end{pmatrix},$$

$$T_2[a] = \begin{pmatrix} S[a] \\ 3 \cdot S[a] \\ 2 \cdot S[a] \\ S[a] \end{pmatrix}, T_3[a] = \begin{pmatrix} S[a] \\ S[a] \\ 3 \cdot S[a] \\ 2 \cdot S[a] \end{pmatrix}$$

AES의 연산인 SubBytes, ShiftRows,
MixColumns를 한 번에 구현 (사전 연산)

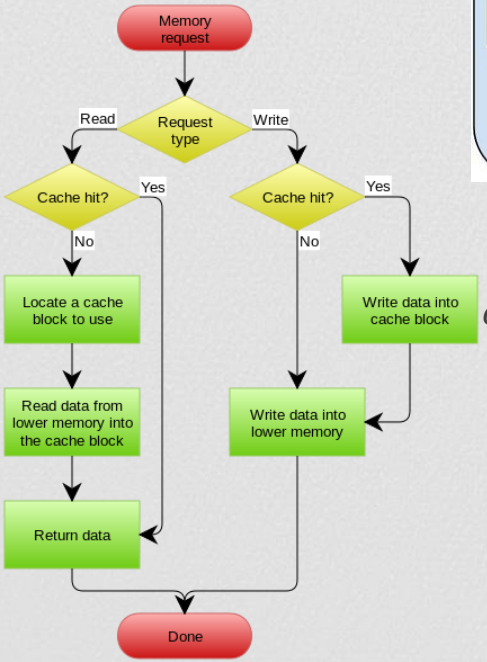
$$\begin{pmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \end{pmatrix} = T_0[a_0, j] \oplus T_1[a_1, j + c_1] \oplus T_2[a_2, j + c_2] \oplus T_3[a_3, j + c_3] \oplus \begin{pmatrix} K_{0,j} \\ K_{1,j} \\ K_{2,j} \\ K_{3,j} \end{pmatrix}$$

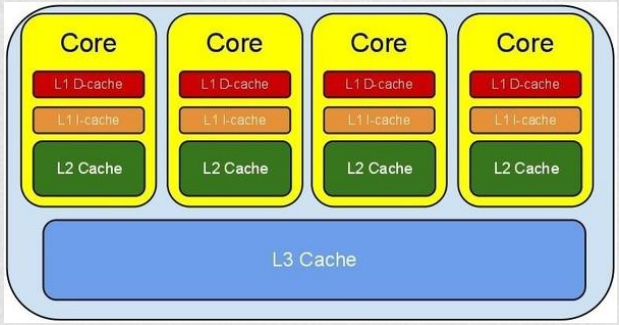
AddRoundKey는 마지막에 연산

# AES 암호화 – T-Table 기반 구현의 문제점

- Cache hit로 인해 발생하는 시간차를 이용하여 공격

캐시의 동작 원리

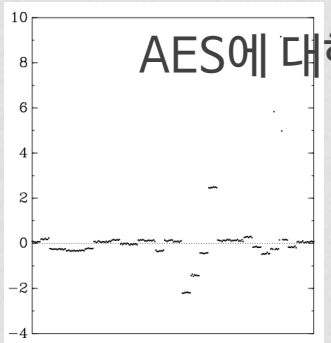



캐시 구조 상세

$$e_j = T_0[a_{0,j}] \oplus T_1[a_{1,j+c_1}] \oplus T_2[a_{2,j+c_2}] \oplus T_3[a_{3,j+c_3}] \oplus K_j$$

AES에 대한 시간차 공격



# AES 암호화 - 비트슬라이싱 구현

## 128-bit 레지스터 8개를 이용한 비트슬라이싱 표현법

- 128-비트 메시지 8개를 병렬 암호화

| | row 3 | | | | | | | ····· | row 0 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | column 0 | | | ··· | column 3 | | | ····· | column 0 | | | ··· | column 3 | | |
| | block 0 | ··· | block 7 | ··· | block 0 | ··· | block 7 | ····· | block 0 | ··· | block 7 | ··· | block 0 | ··· | block 7 |
| $R_0$ | $b_{24}^0$ | ··· | $b_{24}^7$ | ··· | $b_{120}^0$ | ··· | $b_{120}^7$ | ····· | $b_0^0$ | ··· | $b_0^7$ | ··· | $b_{96}^0$ | ··· | $b_{96}^7$ |
| $R_1$ | $b_{25}^0$ | ··· | $b_{25}^7$ | ··· | $b_{121}^0$ | ··· | $b_{121}^7$ | ····· | $b_1^0$ | ··· | $b_1^7$ | ··· | $b_{97}^0$ | ··· | $b_{97}^7$ |
| $R_2$ | $b_{26}^0$ | ··· | $b_{26}^7$ | ··· | $b_{122}^0$ | ··· | $b_{122}^7$ | ····· | $b_2^0$ | ··· | $b_2^7$ | ··· | $b_{98}^0$ | ··· | $b_{98}^7$ |
| $R_3$ | $b_{27}^0$ | ··· | $b_{27}^7$ | ··· | $b_{123}^0$ | ··· | $b_{123}^7$ | ····· | $b_3^0$ | ··· | $b_3^7$ | ··· | $b_{99}^0$ | ··· | $b_{99}^7$ |
| $R_4$ | $b_{28}^0$ | ··· | $b_{28}^7$ | ··· | $b_{124}^0$ | ··· | $b_{124}^7$ | ····· | $b_4^0$ | ··· | $b_4^7$ | ··· | $b_{100}^0$ | ··· | $b_{100}^7$ |
| $R_5$ | $b_{29}^0$ | ··· | $b_{29}^7$ | ··· | $b_{125}^0$ | ··· | $b_{125}^7$ | ····· | $b_5^0$ | ··· | $b_5^7$ | ··· | $b_{101}^0$ | ··· | $b_{101}^7$ |
| $R_6$ | $b_{30}^0$ | ··· | $b_{30}^7$ | ··· | $b_{126}^0$ | ··· | $b_{126}^7$ | ····· | $b_6^0$ | ··· | $b_6^7$ | ··· | $b_{102}^0$ | ··· | $b_{102}^7$ |
| $R_7$ | $b_{31}^0$ | ··· | $b_{31}^7$ | ··· | $b_{127}^0$ | ··· | $b_{127}^7$ | ····· | $b_7^0$ | ··· | $b_7^7$ | ··· | $b_{103}^0$ | ··· | $b_{103}^7$ |

Käsper, E., & Schwabe, P. (2009, September). Faster and timing-attack resistant AES-GCM. In *International Workshop on Cryptographic Hardware and Embedded Systems* (pp. 1-17). Springer, Berlin, Heidelberg.

# AES 암호화 - 비트슬라이싱 구현

## 32-bit 레지스터 8개를 이용한 비트슬라이싱 표현법

- 128-비트 메시지 2개를 병렬 암호화

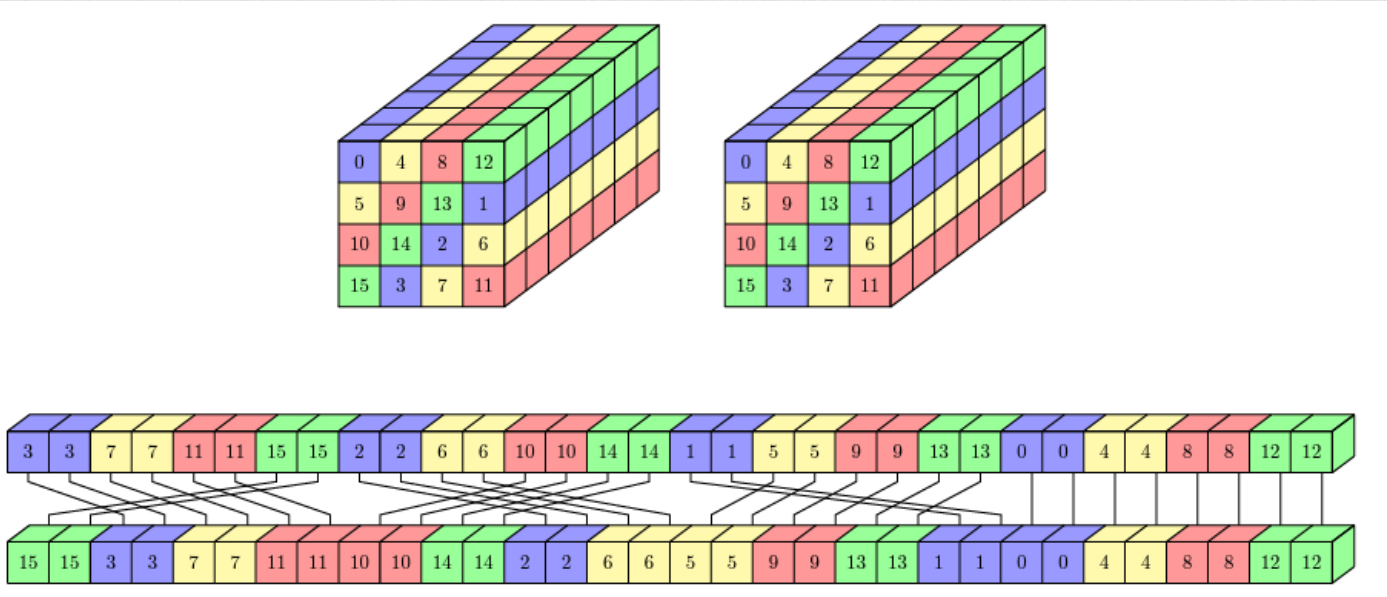
| | row 3 | | | | | | | | ····· | row 0 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | column 0 | | column 1 | | column 2 | | column 3 | | ····· | column 0 | | column 1 | | column 2 | | column 3 | | |
| | block 0 | block 1 | block 0 | block 1 | block 0 | block 1 | block 0 | block 1 | ····· | block 0 | block 1 | block 0 | block 1 | block 0 | block 1 | block 0 | block 1 | |
| $R_0$ | $b_{24}^0$ | $b_{24}^1$ | $b_{56}^0$ | $b_{56}^1$ | $b_{88}^0$ | $b_{88}^1$ | $b_{120}^0$ | $b_{120}^1$ | ····· | $b_0^0$ | $b_0^1$ | $b_{32}^0$ | $b_{32}^1$ | $b_{64}^0$ | $b_{64}^1$ | $b_{96}^0$ | $b_{96}^1$ | |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ····· | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | |
| $R_7$ | $b_{31}^0$ | $b_{31}^1$ | $b_{63}^0$ | $b_{63}^1$ | $b_{95}^0$ | $b_{95}^1$ | $b_{127}^0$ | $b_{127}^1$ | ····· | $b_7^0$ | $b_7^1$ | $b_{39}^0$ | $b_{39}^1$ | $b_{71}^0$ | $b_{71}^1$ | $b_{103}^0$ | $b_{103}^1$ | |

Schwabe, P., & Stoffelen, K. (2016, August). All the AES you need on Cortex-M3 and M4. In *International Conference on Selected Areas in Cryptography* (pp. 180-194). Springer, Cham.

# AES 암호화 - 비트슬라이싱 구현

## 32-bit 레지스터 8개를 이용한 비트슬라이싱과 ShiftRows

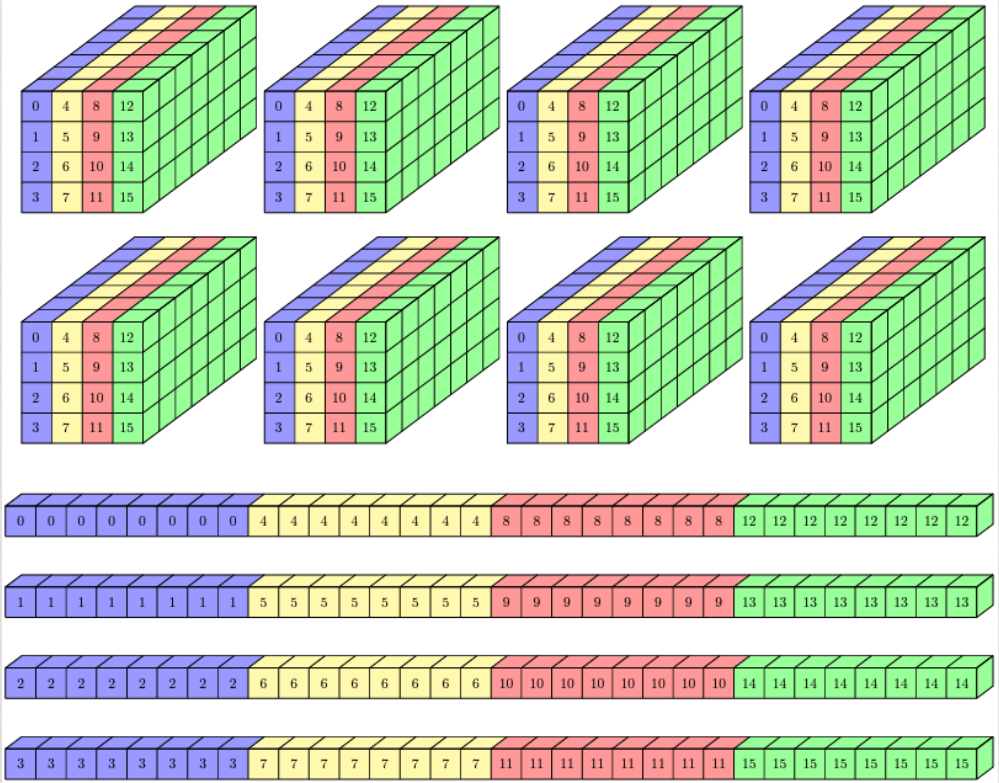
- 완전한 비트슬라이싱이 아니기 때문에 Shift 연산 필요

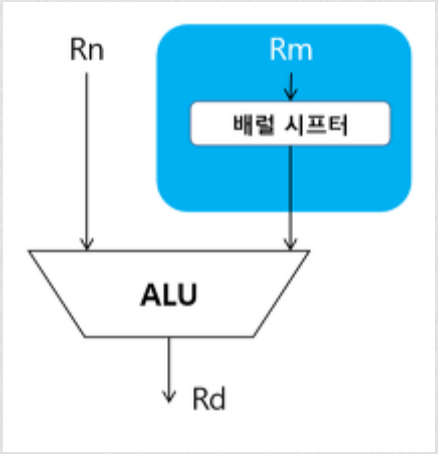


# AES 암호화 - 비트슬라이싱 구현

## 8개의 암호화 블락을 한번에 계산하는 Barrel ShiftRows 표기법

- ShiftRow 연산을 모두 BarrelShifter로 구현 가능

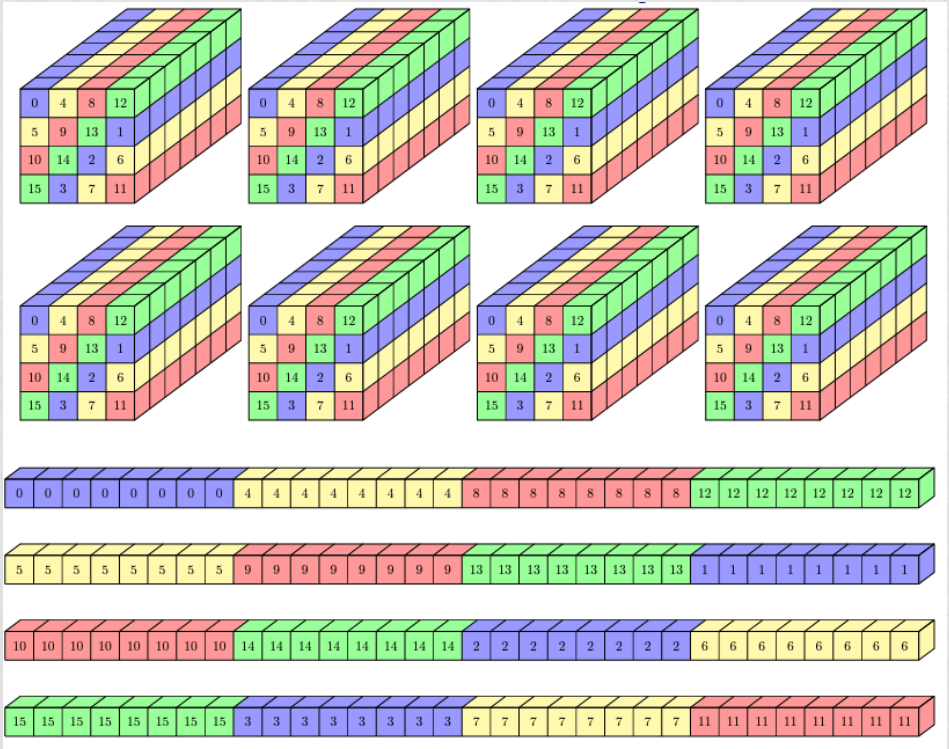


# AES 암호화 - 비트슬라이싱 구현

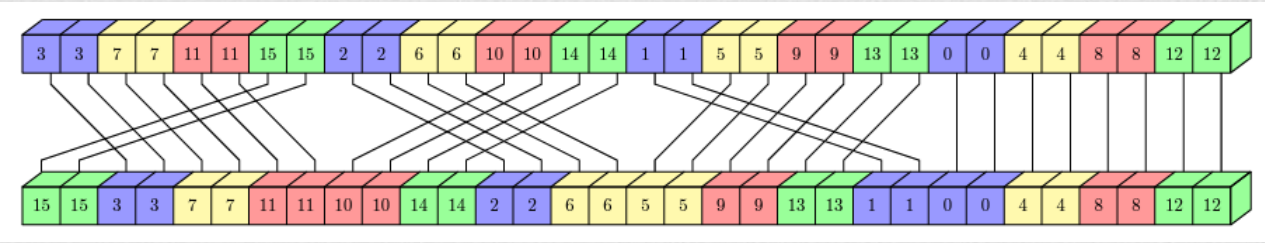
## 8개의 암호화 블락을 한번에 계산하는 Barrel ShiftRows 표기법

- ShiftRow 연산을 모두 BarrelShifter로 구현 가능

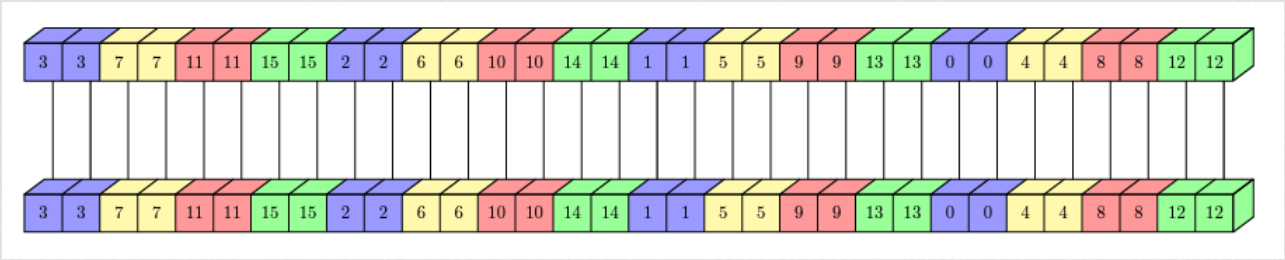



다만 8개의 블락을
한번에 연산해야 함
→ 레지스터가 부족

# AES 암호화 - 픽스슬라이싱 구현




비트슬라이싱 표현법 기반 ShiftRows




픽스슬라이싱 표현법 기반 ShiftRows

**Mixcolumn의 경우 픽스슬라이싱에 특화된 연산 필요**

# AES 암호화 - 픽스슬라이싱 구현

## Mixcolumn 연산

$$a \cdot 02 = (a \ll 1) \oplus (a \gg 7) \wedge (00011011)_2$$

$$a \cdot 03 = a \cdot 02 \oplus a$$

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} = \begin{bmatrix} a \cdot 02 + b \cdot 03 + c + d \\ b \cdot 02 + c \cdot 03 + d + a \\ c \cdot 02 + d \cdot 03 + a + b \\ d \cdot 02 + a \cdot 03 + b + c \end{bmatrix}$$


$$R'_0 = R_1$$
$$R'_1 = R_2$$
$$R'_2 = R_3$$
$$R'_3 = R_4 \oplus R_0$$
$$R'_4 = R_5 \oplus R_0$$
$$R'_5 = R_6$$
$$R'_6 = R_7 \oplus R_0$$
$$R'_7 = R_0$$

# AES 암호화 - 픽스슬라이싱 구현

## Mixcolumn 연산

$$a \cdot 02 = (a \ll 1) \oplus (a \gg 7) \wedge (00011011)_2$$

$$a \cdot 03 = a \cdot 02 \oplus a$$

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} = \begin{bmatrix} a \cdot 02 + b \cdot 03 + c + d \\ b \cdot 02 + c \cdot 03 + d + a \\ c \cdot 02 + d \cdot 03 + a + b \\ d \cdot 02 + a \cdot 03 + b + c \end{bmatrix}$$


$$R'_0 = R_1 \oplus R_1^{\ggg 8} \oplus R_0^{\ggg 8}$$

$$R'_1 = R_2 \oplus R_2^{\ggg 8} \oplus R_1^{\ggg 8}$$

$$R'_2 = R_3 \oplus R_3^{\ggg 8} \oplus R_2^{\ggg 8}$$

$$R'_3 = R_4 \oplus R_0 \oplus R_4^{\ggg 8} \oplus R_3^{\ggg 8} \oplus R_0^{\ggg 8}$$

$$R'_4 = R_5 \oplus R_0 \oplus R_5^{\ggg 8} \oplus R_4^{\ggg 8} \oplus R_0^{\ggg 8}$$

$$R'_5 = R_6 \oplus R_6^{\ggg 8} \oplus R_5^{\ggg 8}$$

$$R'_6 = R_7 \oplus R_0 \oplus R_7^{\ggg 8} \oplus R_6^{\ggg 8} R_0^{\ggg 8}$$

$$R'_7 = R_0 \oplus R_0^{\ggg 8} \oplus R_7^{\ggg 8}$$

# AES 암호화 - 픽스슬라이싱 구현

## Mixcolumn 연산

$$a \cdot 02 = (a \ll 1) \oplus (a \gg 7) \wedge (00011011)_2$$

$$a \cdot 03 = a \cdot 02 \oplus a$$

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} = \begin{bmatrix} a \cdot 02 + b \cdot 03 + c + d \\ b \cdot 02 + c \cdot 03 + d + a \\ c \cdot 02 + d \cdot 03 + a + b \\ d \cdot 02 + a \cdot 03 + b + c \end{bmatrix}$$


$$R'_0 = R_1 \oplus R_1^{\gg 8} \oplus R_0^{\gg 8} \oplus R_0^{\gg 16} \oplus R_0^{\gg 24}$$

$$R'_1 = R_2 \oplus R_2^{\gg 8} \oplus R_1^{\gg 8} \oplus R_1^{\gg 16} \oplus R_1^{\gg 24}$$

$$R'_2 = R_3 \oplus R_3^{\gg 8} \oplus R_2^{\gg 8} \oplus R_2^{\gg 16} \oplus R_2^{\gg 24}$$

$$R'_3 = R_4 \oplus R_0 \oplus R_4^{\gg 8} \oplus R_3^{\gg 8} \oplus R_0^{\gg 8} \oplus R_3^{\gg 16} \oplus R_3^{\gg 24}$$

$$R'_4 = R_5 \oplus R_0 \oplus R_5^{\gg 8} \oplus R_4^{\gg 8} \oplus R_0^{\gg 8} \oplus R_4^{\gg 16} \oplus R_4^{\gg 24}$$

$$R'_5 = R_6 \oplus R_6^{\gg 8} \oplus R_5^{\gg 8} \oplus R_5^{\gg 16} \oplus R_5^{\gg 24}$$

$$R'_6 = R_7 \oplus R_0 \oplus R_7^{\gg 8} \oplus R_6^{\gg 8} R_0^{\gg 8} \oplus R_6^{\gg 16} \oplus R_6^{\gg 24}$$

$$R'_7 = R_0 \oplus R_0^{\gg 8} \oplus R_7^{\gg 8} \oplus R_7^{\gg 16} \oplus R_7^{\gg 24}$$

# AES 암호화 - 픽스슬라이싱 구현

## Mixcolumn 연산

$$a \cdot 02 = (a \ll 1) \oplus (a \gg 7) \wedge (00011011)_2$$

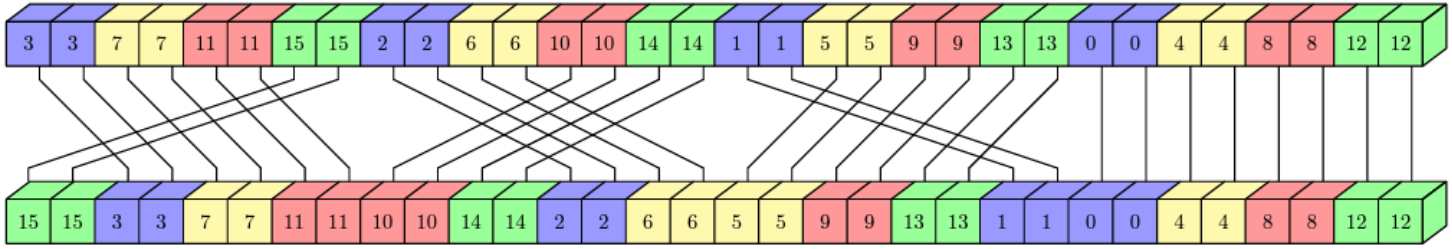
$$a \cdot 03 = a \cdot 02 \oplus a$$

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} = \begin{bmatrix} a \cdot 02 + b \cdot 03 + c + d \\ b \cdot 02 + c \cdot 03 + d + a \\ c \cdot 02 + d \cdot 03 + a + b \\ d \cdot 02 + a \cdot 03 + b + c \end{bmatrix}$$


$$R'_0 = (R_1 \oplus R_1^{\ggg 8}) \oplus R_0^{\ggg 8} \oplus (R_0 \oplus R_0^{\ggg 8})^{\ggg 16}$$
$$R'_1 = (R_2 \oplus R_2^{\ggg 8}) \oplus R_1^{\ggg 8} \oplus (R_1 \oplus R_1^{\ggg 8})^{\ggg 16}$$
$$R'_2 = (R_3 \oplus R_3^{\ggg 8}) \oplus R_2^{\ggg 8} \oplus (R_2 \oplus R_2^{\ggg 8})^{\ggg 16}$$
$$R'_3 = (R_4 \oplus R_4^{\ggg 8}) \oplus R_3^{\ggg 8} \oplus (R_3 \oplus R_3^{\ggg 8})^{\ggg 16} \oplus (R_0 \oplus R_0^{\ggg 8})$$
$$R'_4 = (R_5 \oplus R_5^{\ggg 8}) \oplus R_4^{\ggg 8} \oplus (R_4 \oplus R_4^{\ggg 8})^{\ggg 16} \oplus (R_0 \oplus R_0^{\ggg 8})$$
$$R'_5 = (R_6 \oplus R_6^{\ggg 8}) \oplus R_5^{\ggg 8} \oplus (R_5 \oplus R_5^{\ggg 8})^{\ggg 16}$$
$$R'_6 = (R_7 \oplus R_7^{\ggg 8}) \oplus R_6^{\ggg 8} \oplus (R_6 \oplus R_6^{\ggg 8})^{\ggg 16} \oplus (R_0 \oplus R_0^{\ggg 8})$$
$$R'_7 = (R_0 \oplus R_0^{\ggg 8}) \oplus R_7^{\ggg 8} \oplus (R_7 \oplus R_7^{\ggg 8})^{\ggg 16}$$

# AES 암호화 - 픽스슬라이싱 구현

## Mixcolumn 연산




$$R_0' = \left(R_1 \oplus R_1^{\ggg 8}\right) \oplus R_0^{\ggg 8} \oplus \left(R_0 \oplus R_0^{\ggg 8}\right)^{\ggg 16}$$

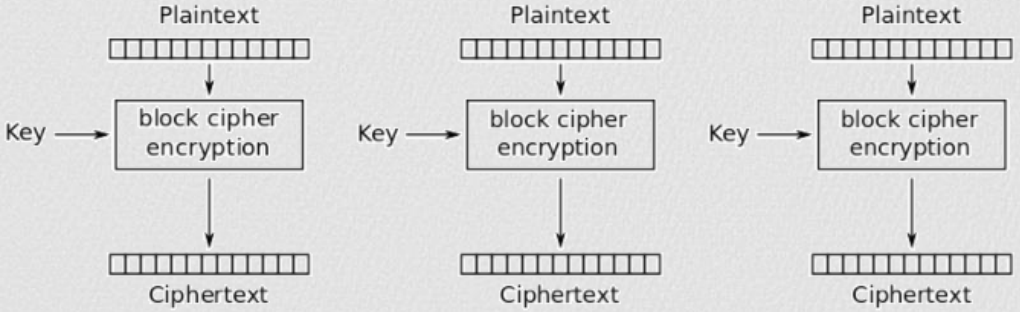
$$R_0' = \left(R_1 \oplus \left(R_1^{\ggg 8} \underset{8}{\ggg} 6\right)\right) \oplus \left(R_0^{\ggg 8} \underset{8}{\ggg} 6\right) \oplus \left(R_0^{\ggg 16} \underset{8}{\ggg} 4\right) \oplus \left(R_0^{\ggg 24} \underset{8}{\ggg} 2\right)$$

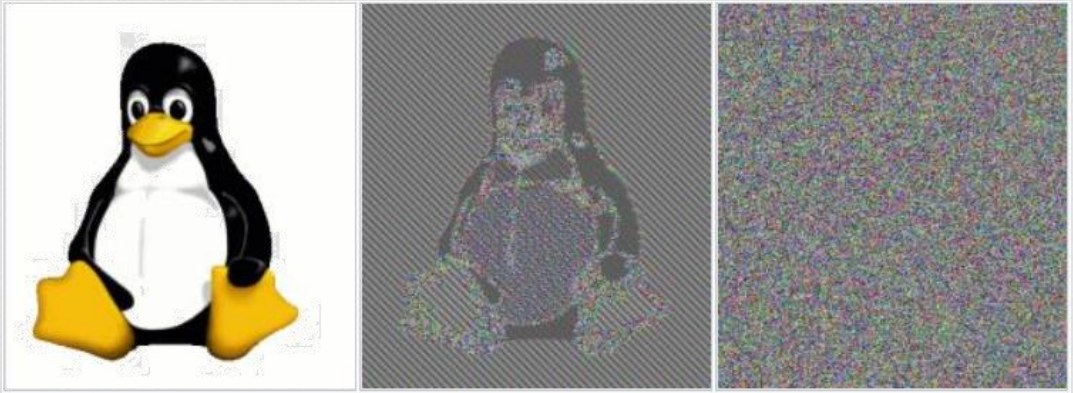
$$R_0' = \left(R_1 \oplus \left(R_1^{\ggg 8} \underset{8}{\ggg} 6\right)\right) \oplus \left(R_0^{\ggg 8} \underset{8}{\ggg} 6\right) \oplus \left(R_0 \oplus \left(R_0^{\ggg 8} \underset{8}{\ggg} 6\right)\right)^{\ggg 16} \underset{8}{\ggg} 4$$

$$R_0' = \left(R_1 \oplus \left(R_1^{\ggg 8} \underset{8}{\ggg} 6\right)\right) \oplus \left(R_0^{\ggg 8} \underset{8}{\ggg} 6\right) \oplus \left(R_0 \oplus \left(R_0^{\ggg 8} \underset{8}{\ggg} 6\right)\right)^{\ggg 16} \underset{8}{\ggg} 4$$

# 운영 모드

## ECB 모드의 문제점



Original image

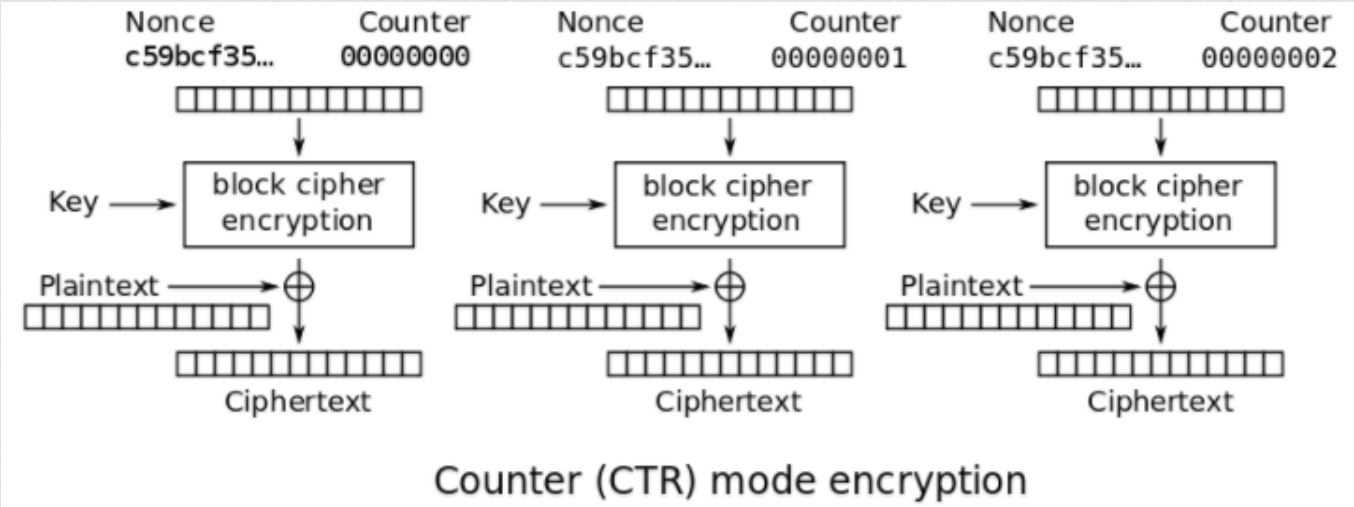
Encrypted using ECB mode

Modes other than ECB result in pseudo-randomness

# 운영 모드

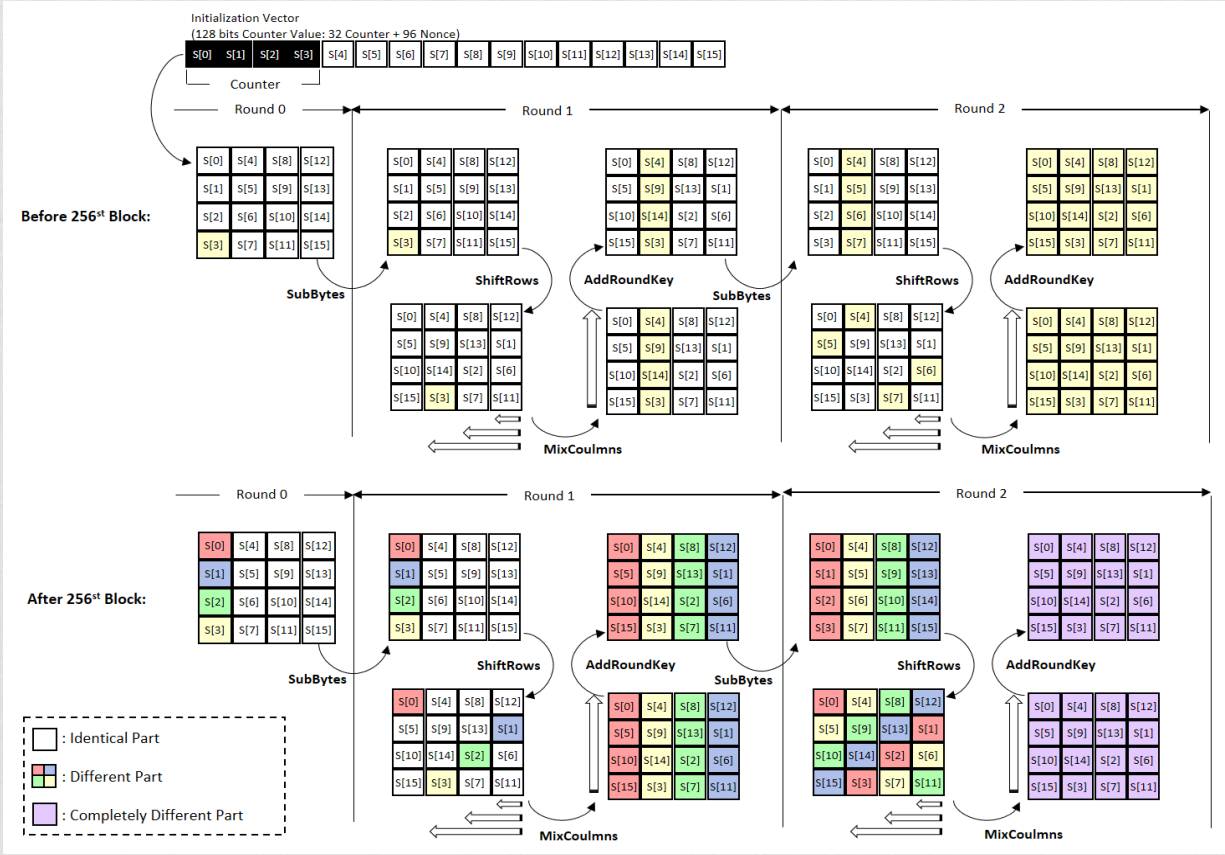
## Counter 모드의 동작 방식

- Counter 값을 1씩 증가 시키며 암호화 수행



# AES-CTR 최적화 구현

## CTR 모드에서 96-비트는 항상 고정→ 사전 계산 가능



# AES-CTR 최적화 구현

## 카운터 값에 따른 전체값 변화 (카운터가 0인 경우)

Initialization Vector
(128 bits Counter Value: 32 Counter + 96 Nonce)

| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Counter

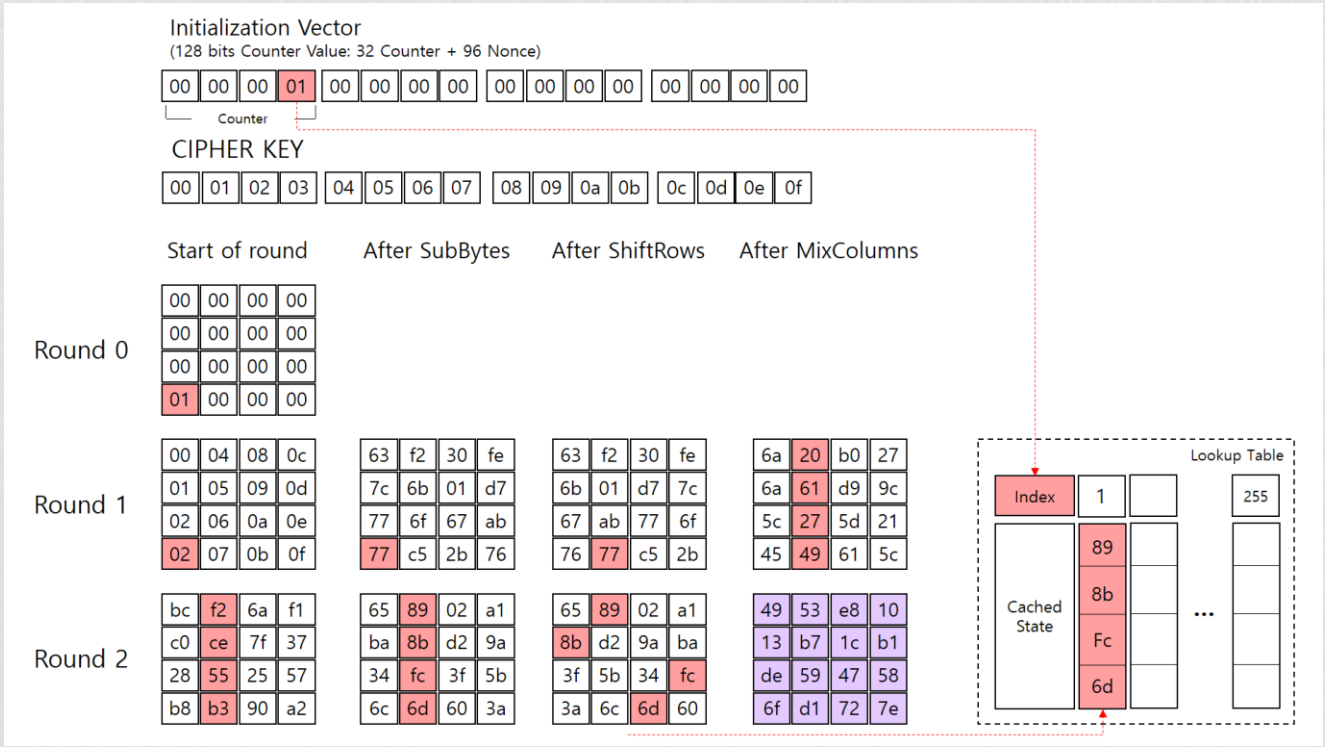
CIPHER KEY

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0a | 0b | 0c | 0d | 0e | 0f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| | Start of round | After SubBytes | After ShiftRows | After MixColumns |
|---|---|---|---|---|
| Round 0 | 00 00 00 00<br>00 00 00 00<br>00 00 00 00<br>00 00 00 00 | | | |
| Round 1 | 00 04 08 0c<br>01 05 09 0d<br>02 06 0a 0e<br>03 07 0b 0f | 63 f2 30 fe<br>7c 6b 01 d7<br>77 6f 67 ab<br>7b c5 2b 76 | 63 f2 30 fe<br>6b 01 d7 7c<br>67 ab 77 6f<br>76 7b c5 2b | 6a 2c b0 27<br>6a 6d d9 9c<br>5c 33 5d 21<br>45 51 61 5c |
| Round 2 | bc fe 6a f1<br>c0 c2 7f 37<br>28 41 25 57<br>b8 ab 90 a2 | 65 bb 02 a1<br>ba 25 d2 9a<br>34 83 3f 5b<br>6c 62 60 3a | 65 bb 02 a1<br>25 d2 9a ba<br>3f 5b 34 83<br>3a 6c 62 60 | a0 37 e7 6f<br>54 85 13 30<br>70 6b 56 a6<br>c1 87 6c 01 |

# AES-CTR 최적화 구현

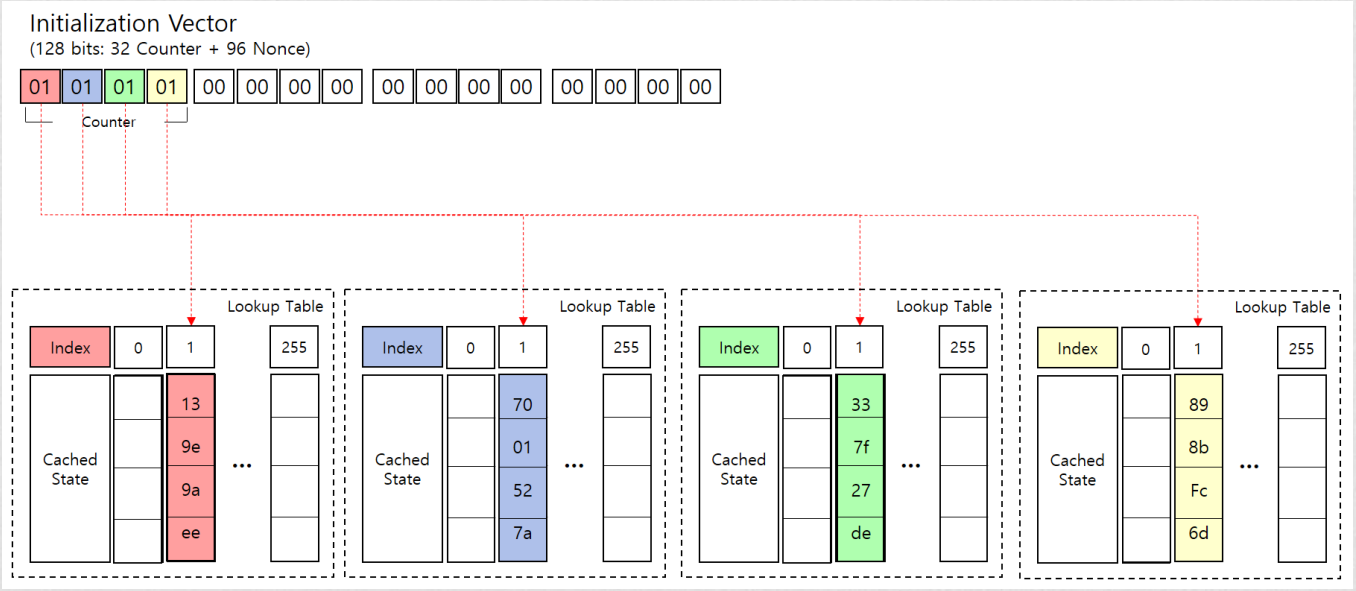
## 카운터 값에 따른 전체값 변화 (카운터가 1인 경우)



# AES-CTR 최적화 구현

## 사전 테이블 구조

- 각 바이트에 해당하는 32-비트 사전 테이블 결과값이 존재



# AES-CTR 최적화 구현

## 최적화 범위

- 2라운드 ShiftRows까지 사전테이블을 통해 연산 가능