SKINNY 블록암호 및 코드분석

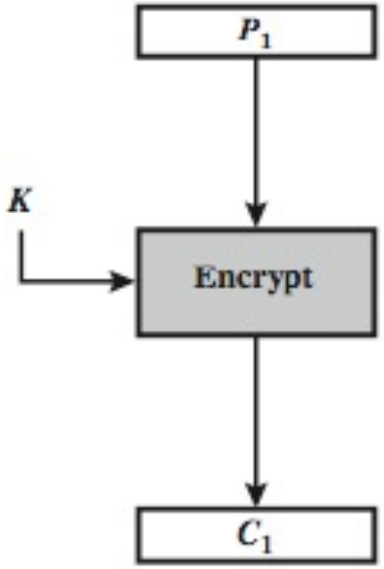
https://youtu.be/m71mZUUxC_0

SKINNY

- CRYPTO 2016에서 발표된 블록 암호
- Tweakable Block cipher (?)
- NIST LWC의 Romulus에서 활용하는 암호
- 암호화 과정은 AES와 비슷한 과정

SKINNY

- Tweakable Block cipher



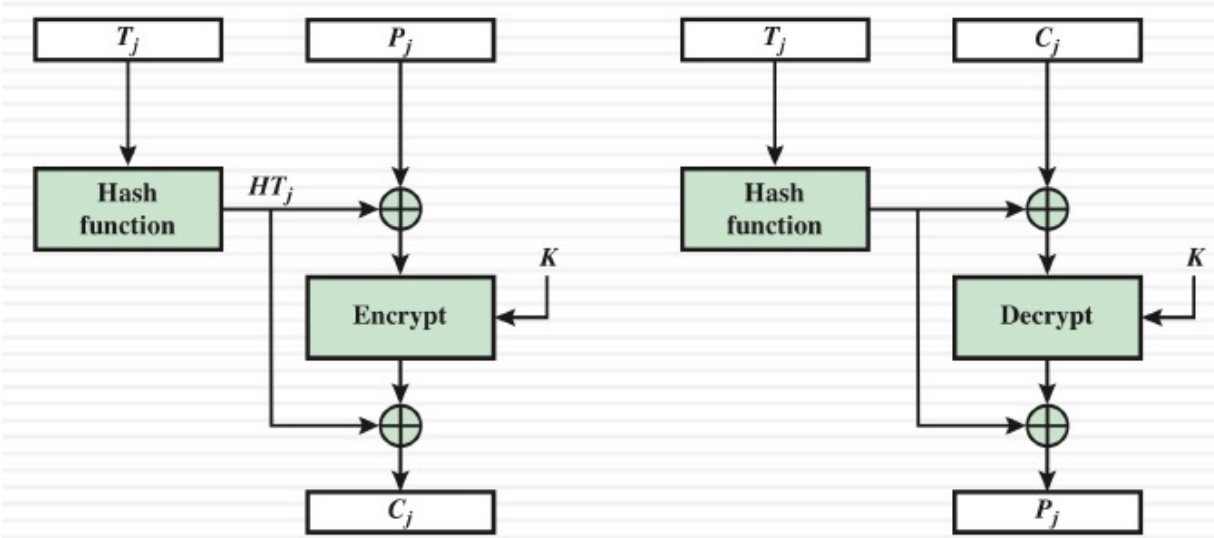


Figure 6.9 Tweakable Block Cipher

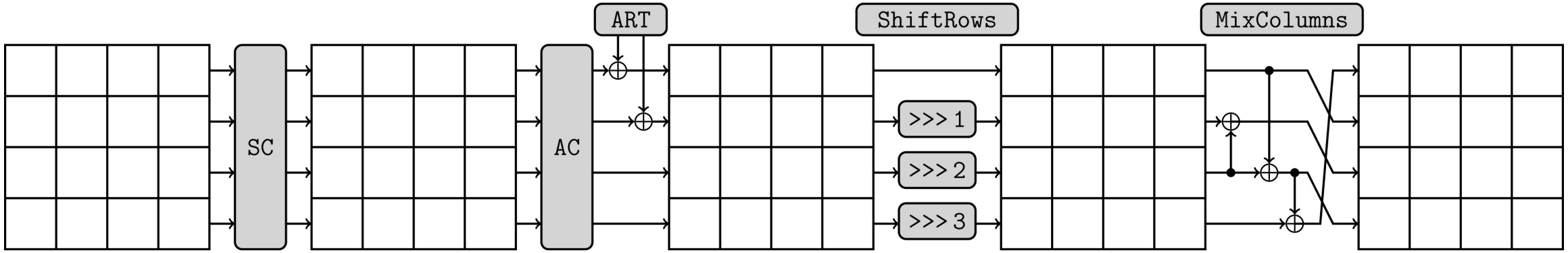
SKINNY

- **Parameters**
  - 블록 길이 (64-bit, 128-bit)
  - 키 길이(t) 는 블록 길이의 x3 까지 지원
  - 64-64/128/192 , 128-128/256/384

	Tweakey size t		
Block size n	n	$2n$	$3n$
64	32 rounds	36 rounds	40 rounds
128	40 rounds	48 rounds	56 rounds

SKINNY

- **The Skinny round function**
  - SubCell
  - AddConstants
  - AddKey
  - ShiftRow
  - MixColumns

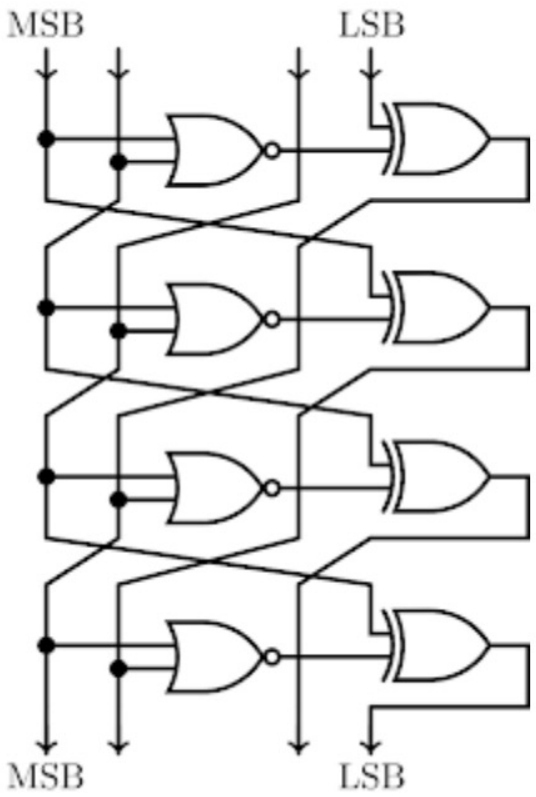


SKINNY - Initialization

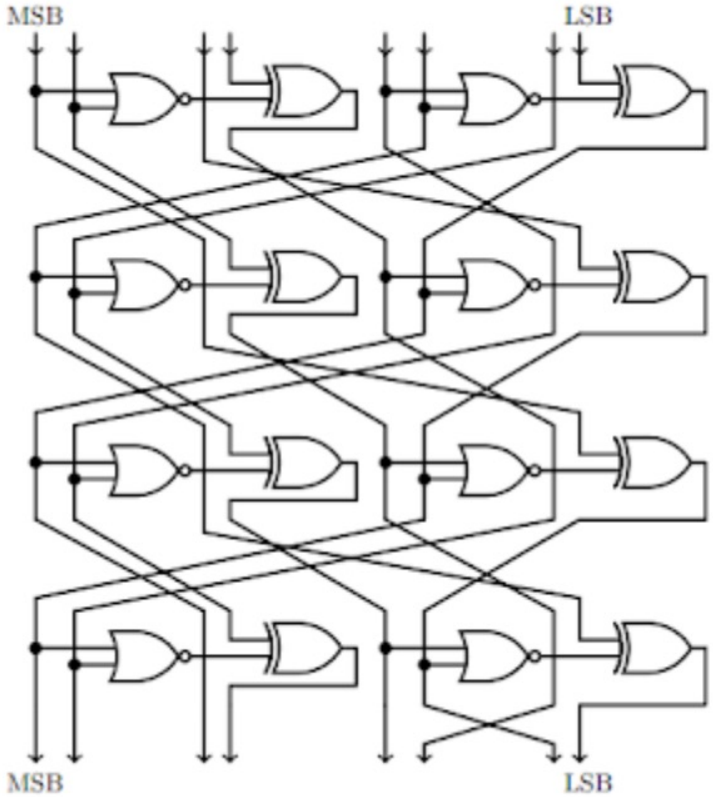
Initialization. The cipher receives a plaintext $m = m_0 \| m_1 \| \cdots \| m_{14} \| m_{15}$, where the m_i are s-bit cells, with $s = n/16$ (we have $s = 4$ for the 64-bit block `SKINNY` versions and $s = 8$ for the 128-bit block `SKINNY` versions). The initialization of the cipher's internal state is performed by simply setting $IS_i = m_i$ for $0 \leq i \leq 15$:

$$IS = \begin{bmatrix} m_0 & m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 & m_7 \\ m_8 & m_9 & m_{10} & m_{11} \\ m_{12} & m_{13} & m_{14} & m_{15} \end{bmatrix}$$

SKINNY - SubCell



The 4-bit Sbox



The 8-bit Sbox

SKINNY - SubCell

```
// 4-bit Sbox
const unsigned char sbox_4[16] = {12,6,9,0,1,10,2,11,3,8,5,13,4,14,7,15};
const unsigned char sbox_4_inv[16] = {3,4,6,8,12,10,1,14,9,2,5,7,0,11,13,15};

// 8-bit Sbox
const unsigned char sbox_8[256] = {0x65 , 0x4c , 0x6a , 0x42 , 0x4b , 0x63 , 0x43 , 0x6b , 0x55 , 0x75 , 0x5a , 0x7a , 0x53 , 0x73 , 0x5b , 0x7b ,0x35 , 0x8c ,
    0x3a , 0x81 , 0x89 , 0x33 , 0x80 , 0x3b , 0x95 , 0x25 , 0x98 , 0x2a , 0x90 , 0x23 , 0x99 , 0x2b ,0xe5 , 0xcc , 0xe8 , 0xc1 , 0xc9 , 0xe0 , 0xc0 , 0xe9 , 0xd5
    , 0xf5 , 0xd8 , 0xf8 , 0xd0 , 0xf0 , 0xd9 , 0xf9 ,0xa5 , 0x1c , 0xa8 , 0x12 , 0x1b , 0xa0 , 0x13 , 0xa9 , 0x05 , 0xb5 , 0x0a , 0xb8 , 0x03 , 0xb0 , 0x0b ,
    0xb9 ,0x32 , 0x88 , 0x3c , 0x85 , 0x8d , 0x34 , 0x84 , 0x3d , 0x91 , 0x22 , 0x9c , 0x2c , 0x94 , 0x24 , 0x9d , 0x2d ,0x62 , 0x4a , 0x6c , 0x45 , 0x4d , 0x64 ,
    0x44 , 0x6d , 0x52 , 0x72 , 0x5c , 0x7c , 0x54 , 0x74 , 0x5d , 0x7d ,0xa1 , 0x1a , 0xac , 0x15 , 0x1d , 0xa4 , 0x14 , 0xad , 0x02 , 0xb1 , 0x0c , 0xbc , 0x04
    , 0xb4 , 0x0d , 0xbd ,0xe1 , 0xc8 , 0xec , 0xc5 , 0xcd , 0xe4 , 0xc4 , 0xed , 0xd1 , 0xf1 , 0xdc , 0xfc , 0xd4 , 0xf4 , 0xdd , 0xfd ,0x36 , 0x8e , 0x38 , 0x82
    , 0x8b , 0x30 , 0x83 , 0x39 , 0x96 , 0x26 , 0x9a , 0x28 , 0x93 , 0x20 , 0x9b , 0x29 ,0x66 , 0x4e , 0x68 , 0x41 , 0x49 , 0x60 , 0x40 , 0x69 , 0x56 , 0x76 ,
    0x58 , 0x78 , 0x50 , 0x70 , 0x59 , 0x79 ,0xa6 , 0x1e , 0xaa , 0x11 , 0x19 , 0xa3 , 0x10 , 0xab , 0x06 , 0xb6 , 0x08 , 0xba , 0x00 , 0xb3 , 0x09 , 0xbb ,0xe6 ,
    0xce , 0xea , 0xc2 , 0xcb , 0xe3 , 0xc3 , 0xeb , 0xd6 , 0xf6 , 0xda , 0xfa , 0xd3 , 0xf3 , 0xdb , 0xfb ,0x31 , 0x8a , 0x3e , 0x86 , 0x8f , 0x37 , 0x87 , 0x3f
    , 0x92 , 0x21 , 0x9e , 0x2e , 0x97 , 0x27 , 0x9f , 0x2f ,0x61 , 0x48 , 0x6e , 0x46 , 0x4f , 0x67 , 0x47 , 0x6f , 0x51 , 0x71 , 0x5e , 0x7e , 0x57 , 0x77 ,
    0x5f , 0x7f ,0xa2 , 0x18 , 0xae , 0x16 , 0x1f , 0xa7 , 0x17 , 0xaf , 0x01 , 0xb2 , 0x0e , 0xbe , 0x07 , 0xb7 , 0x0f , 0xbf ,0xe2 , 0xca , 0xee , 0xc6 , 0xcf ,
    0xe7 , 0xc7 , 0xef , 0xd2 , 0xf2 , 0xde , 0xfe , 0xd7 , 0xf7 , 0xdf , 0xff};
const unsigned char sbox_8_inv[256] = {0xac , 0xe8 , 0x68 , 0x3c , 0x6c , 0x38 , 0xa8 , 0xec , 0xaa , 0xae , 0x3a , 0x3e , 0x6a , 0x6e , 0xea , 0xee ,0xa6 , 0xa3
    , 0x33 , 0x36 , 0x66 , 0x63 , 0xe3 , 0xe6 , 0xe1 , 0xa4 , 0x61 , 0x34 , 0x31 , 0x64 , 0xa1 , 0xe4 ,0x8d , 0xc9 , 0x49 , 0x1d , 0x4d , 0x19 , 0x89 , 0xcd ,
    0x8b , 0x8f , 0x1b , 0x1f , 0x4b , 0x4f , 0xcb , 0xcf ,0x85 , 0xc0 , 0x40 , 0x15 , 0x45 , 0x10 , 0x80 , 0xc5 , 0x82 , 0x87 , 0x12 , 0x17 , 0x42 , 0x47 , 0xc2
    , 0xc7 ,0x96 , 0x93 , 0x03 , 0x06 , 0x56 , 0x53 , 0xd3 , 0xd6 , 0xd1 , 0x94 , 0x51 , 0x04 , 0x01 , 0x54 , 0x91 , 0xd4 ,0x9c , 0xd8 , 0x58 , 0x0c , 0x5c , 0x08
    , 0x98 , 0xdc , 0x9a , 0x9e , 0x0a , 0x0e , 0x5a , 0x5e , 0xda , 0xde ,0x95 , 0xd0 , 0x50 , 0x05 , 0x55 , 0x00 , 0x90 , 0xd5 , 0x92 , 0x97 , 0x02 , 0x07 ,
    0x52 , 0x57 , 0xd2 , 0xd7 ,0x9d , 0xd9 , 0x59 , 0x0d , 0x5d , 0x09 , 0x99 , 0xdd , 0x9b , 0x9f , 0x0b , 0x0f , 0x5b , 0x5f , 0xdb , 0xdf ,0x16 , 0x13 , 0x83 ,
    0x86 , 0x46 , 0x43 , 0xc3 , 0xc6 , 0x41 , 0x14 , 0xc1 , 0x84 , 0x11 , 0x44 , 0x81 , 0xc4 ,0x1c , 0x48 , 0xc8 , 0x8c , 0x4c , 0x18 , 0x88 , 0xcc , 0x1a , 0x1e
    , 0x8a , 0x8e , 0x4a , 0x4e , 0xca , 0xce ,0x35 , 0x60 , 0xe0 , 0xa5 , 0x65 , 0x30 , 0xa0 , 0xe5 , 0x32 , 0x37 , 0xa2 , 0xa7 , 0x62 , 0x67 , 0xe2 , 0xe7 ,0x3d
    , 0x69 , 0xe9 , 0xad , 0x6d , 0x39 , 0xa9 , 0xed , 0x3b , 0x3f , 0xab , 0xaf , 0x6b , 0x6f , 0xeb , 0xef ,0x26 , 0x23 , 0xb3 , 0xb6 , 0x76 , 0x73 , 0xf3 ,
    0xf6 , 0x71 , 0x24 , 0xf1 , 0xb4 , 0x21 , 0x74 , 0xb1 , 0xf4 ,0x2c , 0x78 , 0xf8 , 0xbc , 0x7c , 0x28 , 0xb8 , 0xfc , 0x2a , 0x2e , 0xba , 0xbe , 0x7a , 0x7e
    , 0xfa , 0xfe ,0x25 , 0x70 , 0xf0 , 0xb5 , 0x75 , 0x20 , 0xb0 , 0xf5 , 0x22 , 0x27 , 0xb2 , 0xb7 , 0x72 , 0x77 , 0xf2 , 0xf7 ,0x2d , 0x79 , 0xf9 , 0xbd , 0x7d
    , 0x29 , 0xb9 , 0xfd , 0x2b , 0x2f , 0xbb , 0xbf , 0x7b , 0x7f , 0xfb , 0xff};
```

```
// apply the 4-bit Sbox
void SubCell4(unsigned char state[4][4])
{
    int i,j;
    for(i = 0; i < 4; i++)
        for(j = 0; j <  4; j++)
            state[i][j] = sbox_4[state[i][j]];
}
```

```
// apply the 8-bit Sbox
void SubCell8(unsigned char state[4][4])
{
    int i,j;
    for(i = 0; i < 4; i++)
        for(j = 0; j <  4; j++)
            state[i][j] = sbox_8[state[i][j]];
}
```

SKINNY - AddConstant

$$\begin{bmatrix} c_0 & 0 & 0 & 0 \\ c_1 & 0 & 0 & 0 \\ c_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$(c_0, c_1) = (\mathrm{rc}_3 \| \mathrm{rc}_2 \| \mathrm{rc}_1 \| \mathrm{rc}_0,\ 0 \| 0 \| \mathrm{rc}_5 \| \mathrm{rc}_4) \text{ when } s = 4 \qquad c_2 = \texttt{0x2}$$

$$(c_0, c_1) = (0 \| 0 \| 0 \| 0 \| \mathrm{rc}_3 \| \mathrm{rc}_2 \| \mathrm{rc}_1 \| \mathrm{rc}_0,\ 0 \| 0 \| 0 \| 0 \| 0 \| 0 \| \mathrm{rc}_5 \| \mathrm{rc}_4) \text{ when } s = 8.$$

Rounds	Constants
1 - 16	01,03,07,0F,1F,3E,3D,3B,37,2F,1E,3C,39,33,27,0E
17 - 32	1D,3A,35,2B,16,2C,18,30,21,02,05,0B,17,2E,1C,38
33 - 48	31,23,06,0D,1B,36,2D,1A,34,29,12,24,08,11,22,04
49 - 62	09,13,26,0C,19,32,25,0A,15,2A,14,28,10,20

SKINNY - AddConstant

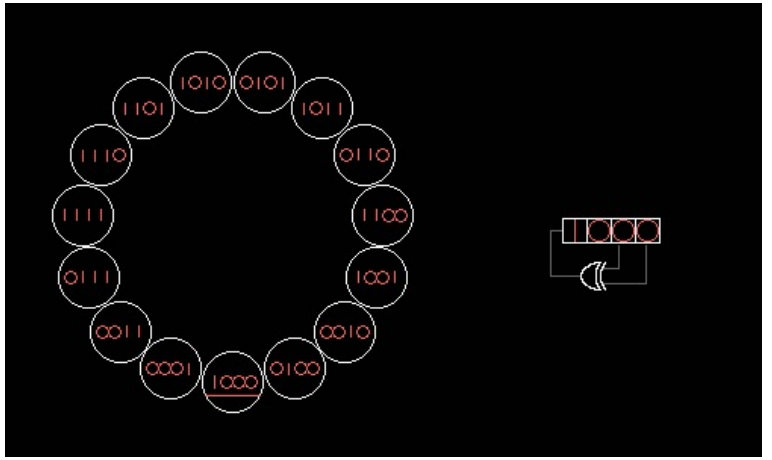
$$(c_0, c_1) = (\mathrm{rc}_3 \| \mathrm{rc}_2 \| \mathrm{rc}_1 \| \mathrm{rc}_0,\ 0 \| 0 \| \mathrm{rc}_5 \| \mathrm{rc}_4) \text{ when } s = 4$$
$$(c_0, c_1) = (0 \| 0 \| 0 \| 0 \| \mathrm{rc}_3 \| \mathrm{rc}_2 \| \mathrm{rc}_1 \| \mathrm{rc}_0,\ 0 \| 0 \| 0 \| 0 \| 0 \| 0 \| \mathrm{rc}_5 \| \mathrm{rc}_4) \text{ when } s = 8.$$

1 - 16 `01,03,07,0F,1F,3E,3D,3B,37,2F,1E,3C,39,33,27,0E`

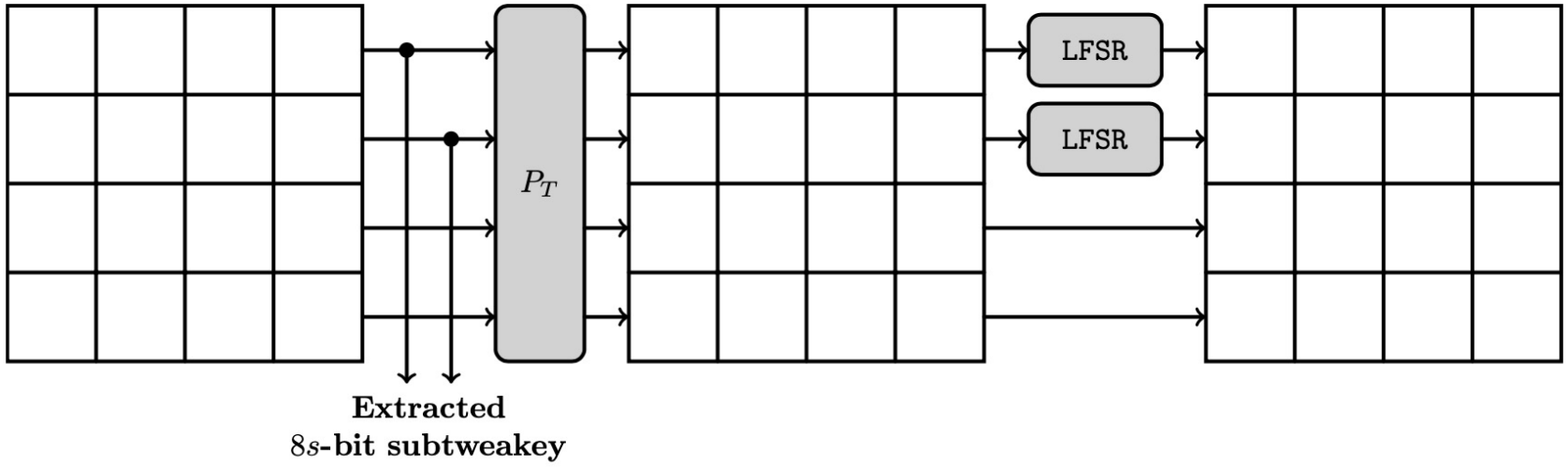
```c
void AddConstants(unsigned char state[4][4], int r)
{
    state[0][0] ^= (RC[r] & 0xf);
    state[1][0] ^= ((RC[r]>>4) & 0x3);
    state[2][0] ^= 0x2;
}
```

SKINNY - AddKey

- LFSR(Linear Feedback Shift Register)



- $IS_{i,j} = IS_{i,j} \oplus TK1_{i,j}$ when $z = 1$,
- $IS_{i,j} = IS_{i,j} \oplus TK1_{i,j} \oplus TK2_{i,j}$ when $z = 2$,
- $IS_{i,j} = IS_{i,j} \oplus TK1_{i,j} \oplus TK2_{i,j} \oplus TK3_{i,j}$ when $z = 3$.



SKINNY - AddKey

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

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

$$(0, \ldots, 15) \xmapsto{P_T} (9, 15, 8, 13, 10, 14, 12, 11, 0, 1, 2, 3, 4, 5, 6, 7)$$

TK	s	**LFSR**
$TK2$	4	$(x_3 \|\| x_2 \|\| x_1 \|\| x_0) \rightarrow (x_2 \|\| x_1 \|\| x_0 \|\| x_3 \oplus x_2)$
	8	$(x_7 \|\| x_6 \|\| x_5 \|\| x_4 \|\| x_3 \|\| x_2 \|\| x_1 \|\| x_0) \rightarrow (x_6 \|\| x_5 \|\| x_4 \|\| x_3 \|\| x_2 \|\| x_1 \|\| x_0 \|\| x_7 \oplus x_5)$
$TK3$	4	$(x_3 \|\| x_2 \|\| x_1 \|\| x_0) \rightarrow (x_0 \oplus x_3 \|\| x_3 \|\| x_2 \|\| x_1)$
	8	$(x_7 \|\| x_6 \|\| x_5 \|\| x_4 \|\| x_3 \|\| x_2 \|\| x_1 \|\| x_0) \rightarrow (x_0 \oplus x_6 \|\| x_7 \|\| x_6 \|\| x_5 \|\| x_4 \|\| x_3 \|\| x_2 \|\| x_1)$

SKINNY - AddKey

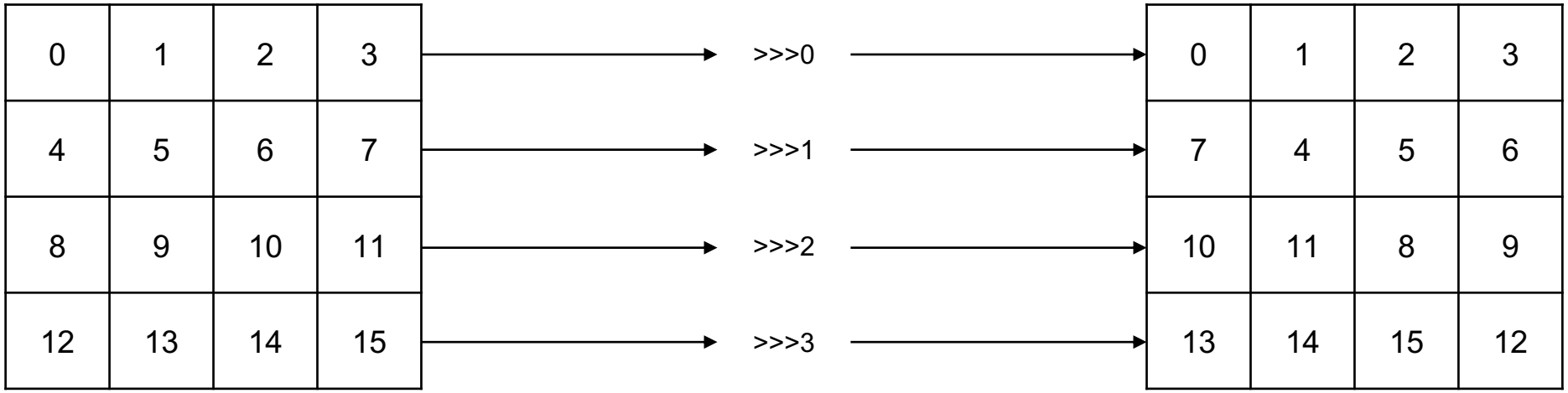
```
unsigned char keyCells_tmp[3][4][4];
```

```
for(i = 0; i <= 1; i++)
{
    for(j = 0; j < 4; j++)
    {
        state[i][j] ^= keyCells[0][i][j];
        if (2*versions[ver][0]==versions[ver][1])
            state[i][j] ^= keyCells[1][i][j];
        else if (3*versions[ver][0]==versions[ver][1])
           state[i][j] ^= keyCells[1][i][j] ^ keyCells[2][i][j];
    }
}
```

```
// update the subtweakey states with the permutation
for(k = 0; k <(int)(versions[ver][1]/versions[ver][0]); k++){
    for(i = 0; i < 4; i++){
        for(j = 0; j < 4; j++){
            //application of the TWEAKEY permutation
            pos=TWEAKEY_P[j+4*i];
            keyCells_tmp[k][i][j]=keyCells[k][pos>>2][pos&0x3];
        }
    }
}
```

SKINNY - ShiftRow

$$P = [0, 1, 2, 3, 7, 4, 5, 6, 10, 11, 8, 9, 13, 14, 15, 12].$$



SKINNY - MixColumns

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \cdot$$

00	01	02	03
10	11	12	13
20	21	22	23
30	31	32	33


```c
void MixColumn(unsigned char state[4][4])
{
    int j;
    unsigned char temp;

    for(j = 0; j < 4; j++){
        state[1][j]^=state[2][j];
        state[2][j]^=state[0][j];
        state[3][j]^=state[2][j];

        temp=state[3][j];
        state[3][j]=state[2][j];
        state[2][j]=state[1][j];
        state[1][j]=state[0][j];
        state[0][j]=temp;
    }
}
```

기 타

```
if(i&1)
{
    state[i>>2][i&0x3] = input[i>>1]&0xF;
    keyCells[0][i>>2][i&0x3] = userkey[i>>1]&0xF;
    if (versions[ver][1]>=128)
        keyCells[1][i>>2][i&0x3] = userkey[(i+16)>>1]&0xF;
    if (versions[ver][1]>=192)
        keyCells[2][i>>2][i&0x3] = userkey[(i+32)>>1]&0xF;
}
else
{
    state[i>>2][i&0x3] = (input[i>>1]>>4)&0xF;
    keyCells[0][i>>2][i&0x3] = (userkey[i>>1]>>4)&0xF;
    if (versions[ver][1]>=128)
        keyCells[1][i>>2][i&0x3] = (userkey[(i+16)>>1]>>4)&0xF;
    if (versions[ver][1]>=192)
        keyCells[2][i>>2][i&0x3] = (userkey[(i+32)>>1]>>4)&0xF;
}
```

Q & A