void expandKey(BYTE \*key, BYTE \*roundKey)

1) key: 키 스케줄링을 수행할 16바이트 키

2) roundKey: 키 스케줄링의 결과인 176바이트 라운드 키가 담길 공간

STEP1 : 128비트의 키를 4개의 32비트 워드로 바꾼다.

STEP2 : 첫 4개의 워드 중 마지막 워드는 1바이트 왼쪽 순환 이동된 뒤 S-Box를 이용하여 치환된다. 그 다음에 라운드 상수와 XOR된다.

STEP3 : 첫 워드와 기존 4개의 워드 중 두번째 워드가 XOR되어 두번째 워드가 생성, 이결과를 반복한다.

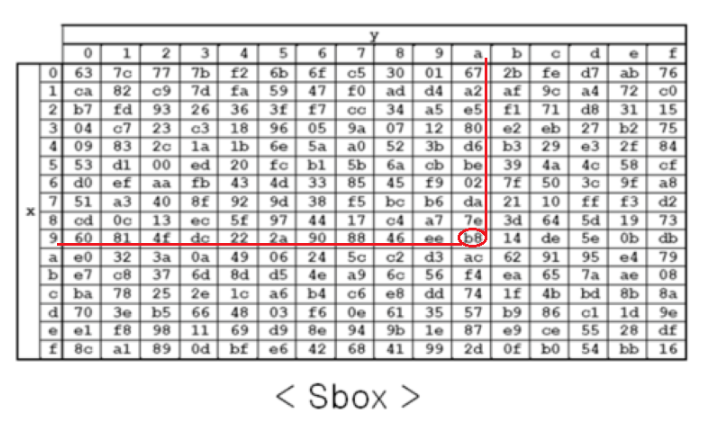
STEP4 : 2~3을 9번 수행하여 라운트 키를 생성한다.

BYTE\* subBytes(BYTE \*block, int mode)

1) block: SubBytes 수행할 16바이트 블록, 수행 결과는 해당 배열에 바로 반영

2) mode: SubByte 수행 모드

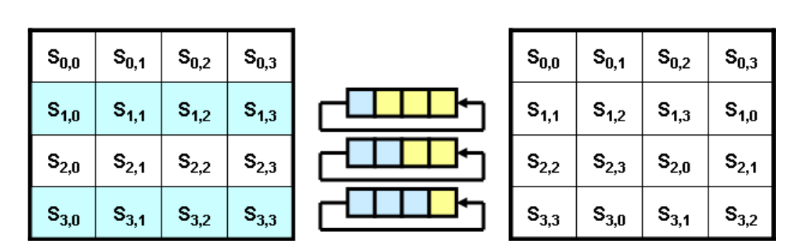
Step1 : AES state를 Sbox통해서 치환



BYTE\* shiftRows(BYTE \*block, int mode)

1) block: ShiftRows 수행할 16바이트 블록, 수행 결과는 해당 배열에 바로 반영

2) mode: ShiftRows수행 모드

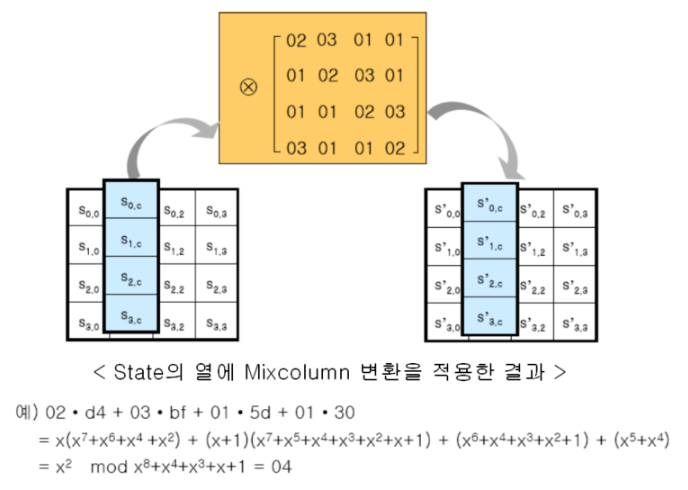


BYTE\* mixColumns(BYTE \*block, int mode)

1) block: MixColumns을 수행할 16바이트 블록, 수행 결과는 해당 배열에 바로 반영

2) mode: MixColumns의 수행 모드

Step : state에 각 Colume에 대해서 아래와 같이 행렬 곱셈을 수행



BYTE\* addRoundKey(BYTE \*block, BYTE \*rKey)

1) block: AddRoundKey를 수행할 16바이트 블록, 수행 결과는 해당 배열에 바로 반영

2) rKey: AddRoundKey를 수행할 16바이트 라운드키

Step : State내의 각각의 byte들이 각 roundkey와 xor연산되어진다

void AES128(BYTE \*input, BYTE \*output, BYTE \*key, int mode)

mode가 ENC일 경우 평문을 암호화하고, DEC일 경우 암호문을 복호화하는 함수

