Chapter 5
Advanced Encryption Standard

정보보안

Abridged version

Origins

- Advanced Encryption Standard (AES) 대칭형.
  - AES is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.
  - 암호 이름 이것으로 이름을 명능 만든 사람
  - Based on the Rijndael cipher developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, who submitted a proposal which was evaluated by the NIST during the AES selection process.
  - AES has been adopted by the U.S. government and is now used worldwide. It supersedes the Data Encryption Standard (DES), which was published in 1977. 약하고 지금도 씀

한국에서 살아남은건 CRYPTON

AES General Structure

- GF(2^8)
  - AES uses arithmetic in GF(2^8) with the irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$. Therefore, $x^8 = x^4 + x^3 + x + 1 = (00011011) = \{1B\}$.

    x^8을 넘김.

    x^8을 치환
  - For $A = (a_7a_6 \ldots a_1a_0)$ and $B = (b_7b_6 \ldots b_1b_0)$,
    the sum is $A + B = (c_7c_6 \ldots c_1c_0)$ where $c_i = a_i \oplus b_i$ and the multiplication $\{02\} \cdot A$ is $(a_6 \ldots a_1a_0 0)$ if $a_7 = 0$ and is $(a_6 \ldots a_1a_0 0) \oplus (00011011)$ if $a_7 = 1$.

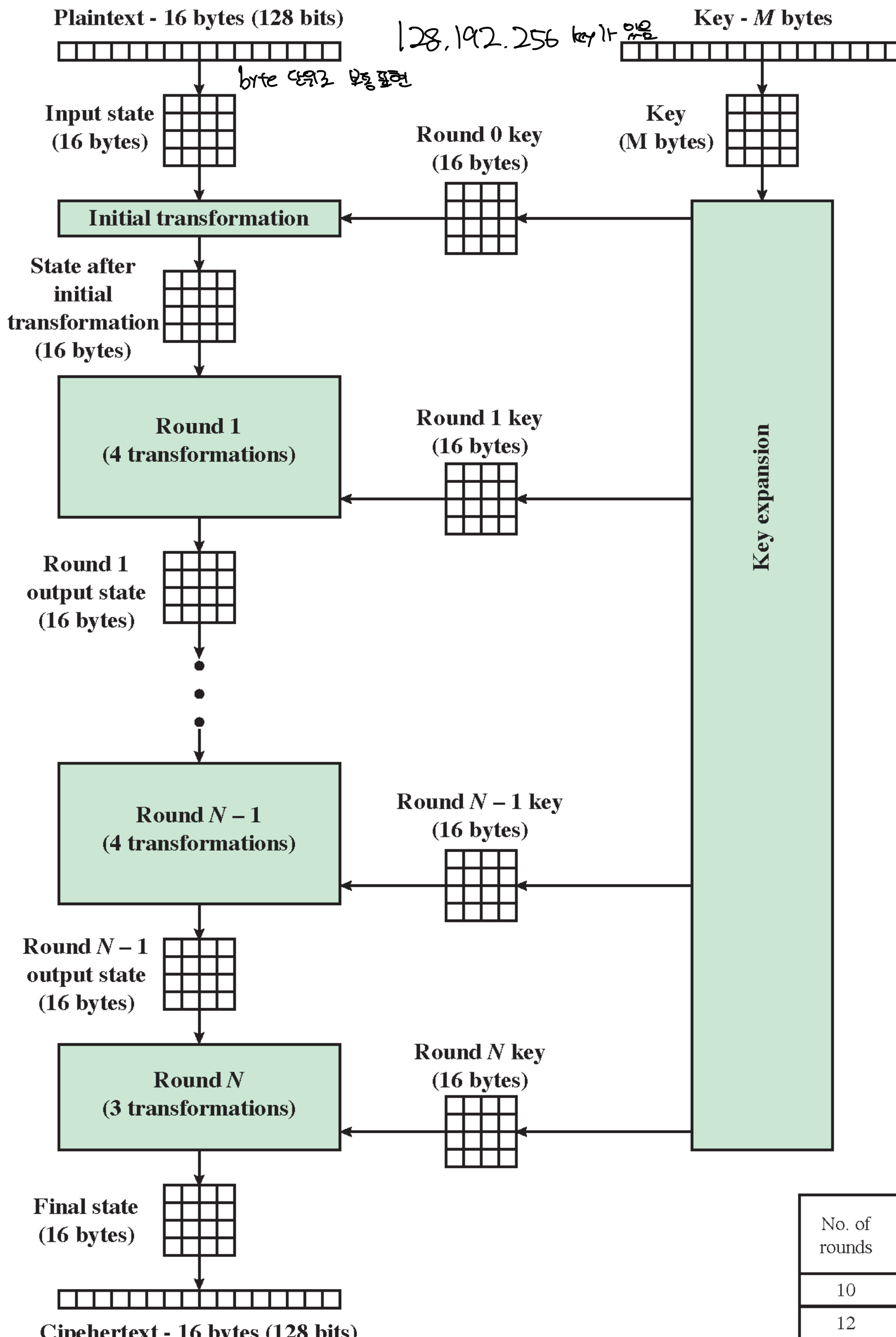
    0000 0010 → x 7차가 0이면 7차가 1이면

- Parameter

  깰려면 2^{128}

  AES : 128 bit로 고정.
  - AES takes a plaintext block size of 128 bits (16 bytes).
  - The key length can be 128, 192, or 256 bits (16, 24, or 32 bytes).
  - The number of rounds is 10, 12, or 14. key는 3개. 128. 192. 256
  - The algorithm is referred to as AES-128, AES-192, or AES-256, depending on the key length.

    2^{128}
- State
  - State is a data block of 4 columns of 4 bytes, which is depicted as 4 × 4 square matrix of bytes.
  - Similarly, the key is depicted as a square matrix of bytes.



No. of rounds	Key Length (bytes)
10	16
12	24
14	32

Figure 5.1 AES Encryption Process

각 Round 마다 transformation 4개
마지막 〃 는 3개, 시작할때 +1개
시작이랑 끝을 보통 다르게 만듦
fiestel 구조는 안씀. 연구가 많았어서 트렌드 지남

AES Detailed Structure

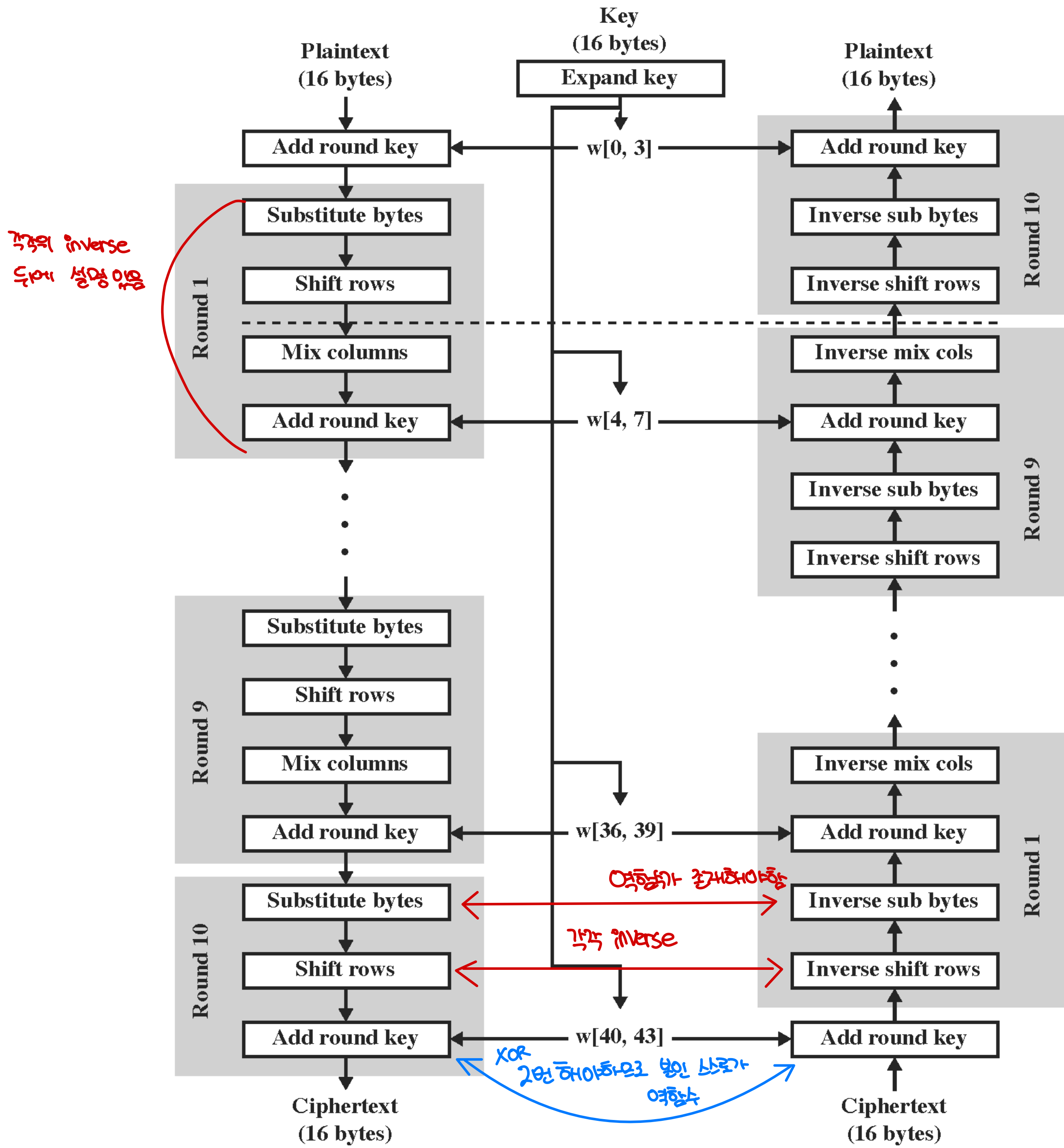
인벌스 필요 X.

- AES is not a Feistel structure. 근데 AES는 인벌스 필요
- Four different (reversible) transformations are used.

역함수 존재해야함

  - SubBytes Byte끼리.
  - ShiftRows Row끼리
  - MixColumns columns끼리
  - AddRoundKey key끼리
- Overall structure
  - AES-128 begins with an AddRoundKey transformation, followed by nine rounds that each includes all four transformations, followed by a tenth round of three transformations.
  - Only the AddRoundKey transformation makes use of the key.

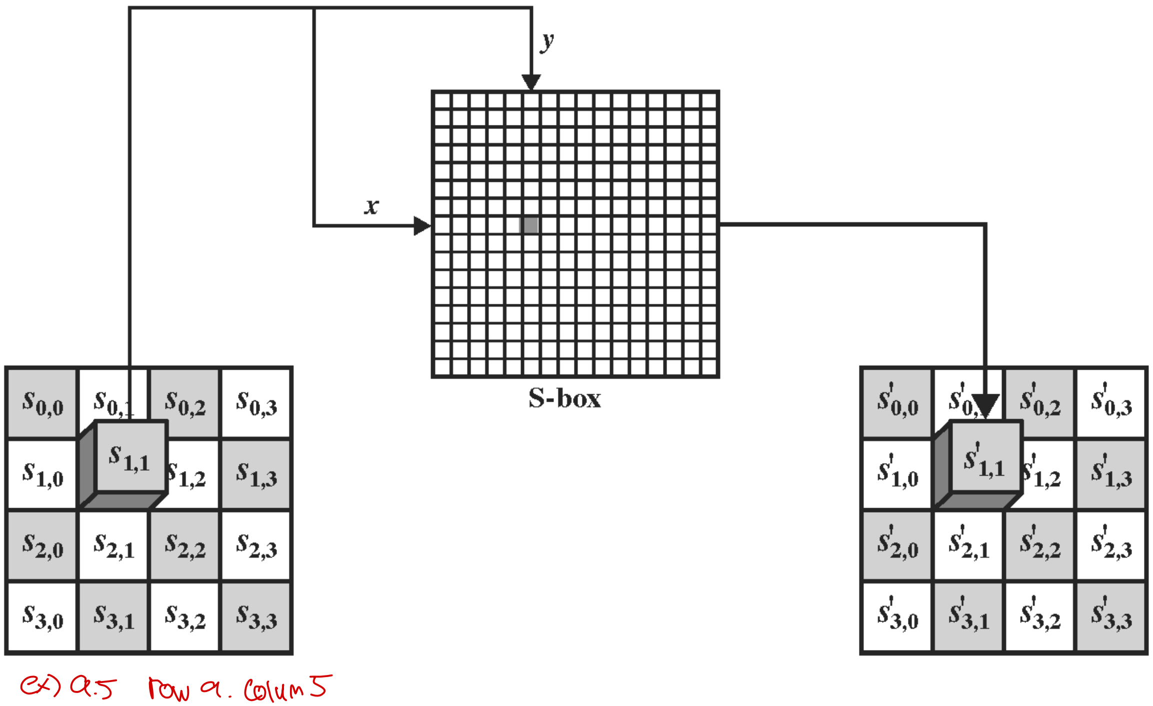
9번 round는 4번 trans, 마지막 R은 3번



(a) Encryption

(b) Decryption

SubBytes Transformation



Substitute byte transformation

- The leftmost 4 bits of the State are used as a row value and the rightmost 4 bits are used as a column value.
- For example, the hexadecimal value {95} references row 9, column 5 of the S-box, which contains the value {2A}. Accordingly, {95} is mapped into {2A}.

Table 5.2 AES S-Boxes

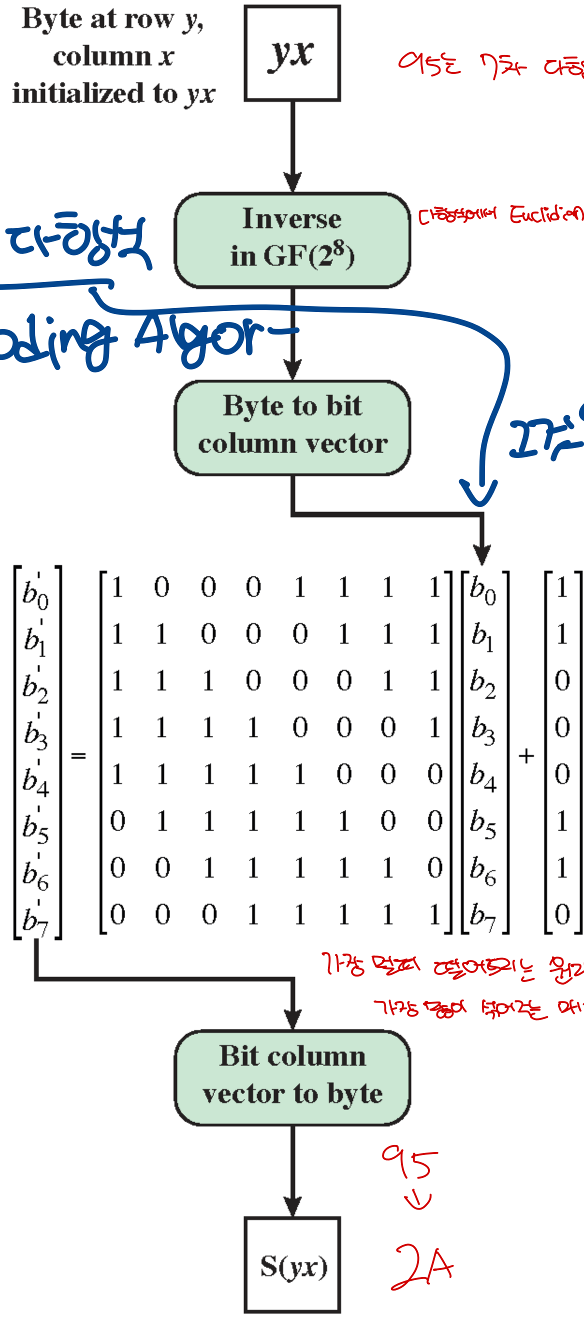
(a) S-box

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

95가 왜 2A로 바뀌는가? 뒷페이지

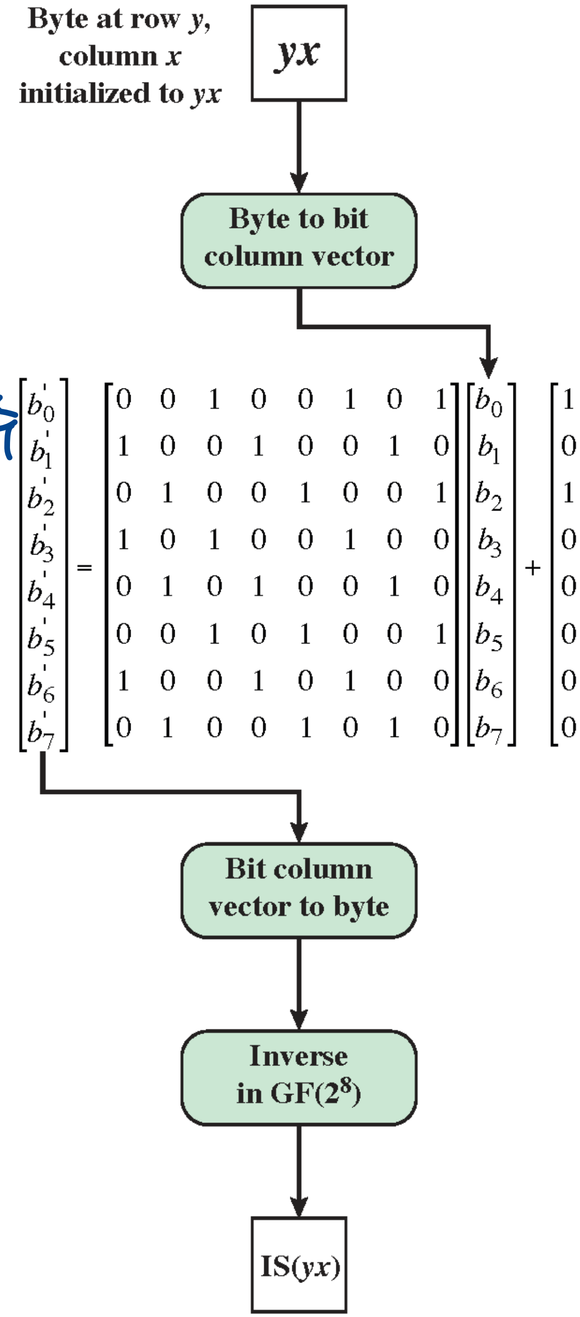
(b) Inverse S-box

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
	2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
	3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
	5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
	6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
	7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
	8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
	9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
	A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
	B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
	C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
	D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
	E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
	F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D



$$\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{bmatrix} = \begin{bmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

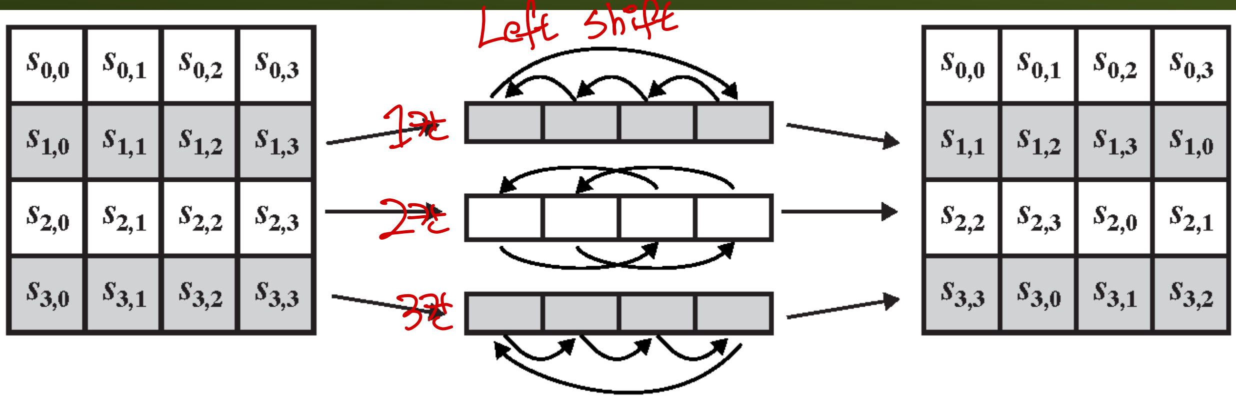
(a) Calculation of byte at row y, column x of S-box



$$\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{bmatrix} = \begin{bmatrix} 0&0&1&0&0&1&0&1 \\ 1&0&0&1&0&0&1&0 \\ 0&1&0&0&1&0&0&1 \\ 1&0&1&0&0&1&0&0 \\ 0&1&0&1&0&0&1&0 \\ 0&0&1&0&1&0&0&1 \\ 1&0&0&1&0&1&0&0 \\ 0&1&0&0&1&0&1&0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

(a) Calculation of byte at row y, column x of IS-box

ShiftRows Transformation



87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

inverse 계산은 오른쪽으로 0.1.2.3 shift

→

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

- The first row of the State is not altered.

  For the 2nd row, a 1-byte circular left shift is performed.

  For the 3rd row, a 2-byte circular left shift is performed.

  For the 4th row, a 3-byte circular left shift is performed.
- The inverse shift row transformation, called InvShiftRows, performs the circular right shirt for each of the last three rows, with a 1-byte circular right shift for the 2nd row, and so on.

MixColumns Transformation

$$\begin{array}{cccc} x & x+1 & 1 & 1 \\ \end{array}$$

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix} \quad \textbf{(5.3)}$$

input output

AES는 Encryption 빠름
decryption 더 느림

효율적

$$s'_{0,j} = (2 \cdot s_{0,j}) \oplus (3 \cdot s_{1,j}) \oplus s_{2,j} \oplus s_{3,j}$$

$$s'_{1,j} = s_{0,j} \oplus (2 \cdot s_{1,j}) \oplus (3 \cdot s_{2,j}) \oplus s_{3,j}$$

$$s'_{2,j} = s_{0,j} \oplus s_{1,j} \oplus (2 \cdot s_{2,j}) \oplus (3 \cdot s_{3,j})$$

$$s'_{3,j} = (3 \cdot s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \cdot s_{3,j})$$

InvMixColumns Transformation

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix} \qquad \textbf{(5.5)}$$

It is not immediately clear that Equation (5.5) is the **inverse** of Equation (5.3). We need to show

각원소별로 Message랑 XOR.

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$

which is equivalent to showing

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \textbf{(5.6)}$$

AddRoundKey Transformation

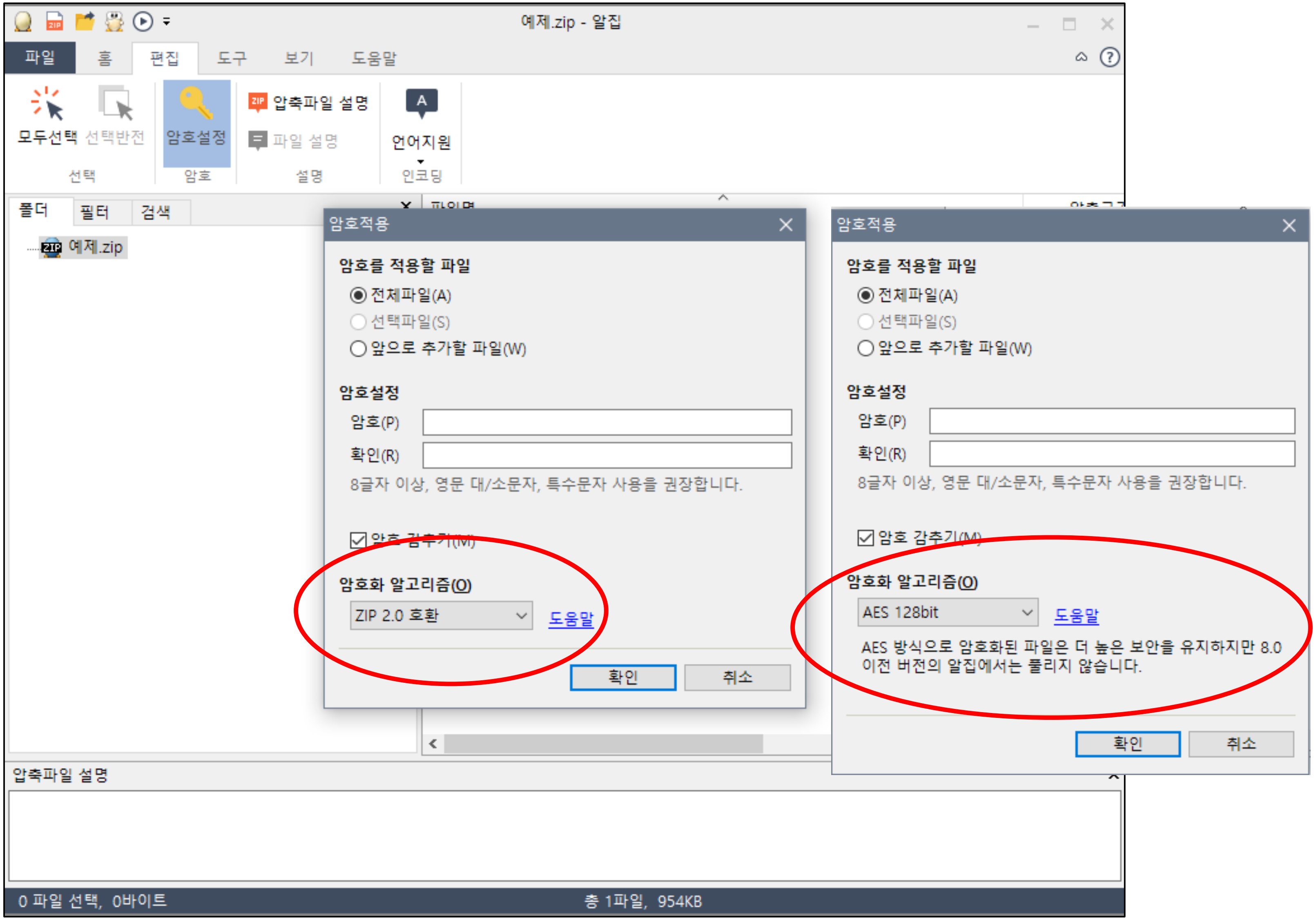
FORWARD AND INVERSE TRANSFORMATIONS In the **forward add round key transformation**, called AddRoundKey, the 128 bits of **State** are bitwise XORed with the 128 bits of the round key.

$$
\begin{bmatrix} 47 & 40 & A3 & 4C \\ 37 & D4 & 70 & 9F \\ 94 & E4 & 3A & 42 \\ ED & A5 & A6 & BC \end{bmatrix}
\oplus
\begin{bmatrix} AC & 19 & 28 & 57 \\ 77 & FA & D1 & 5C \\ 66 & DC & 29 & 00 \\ F3 & 21 & 41 & 6A \end{bmatrix}
=
\begin{bmatrix} EB & 59 & 8B & 1B \\ 40 & 2E & A1 & C3 \\ F2 & 38 & 13 & 42 \\ 1E & 84 & E7 & D6 \end{bmatrix}
$$

The first matrix is **State**, and the second matrix is the round key.

The **inverse add round key transformation** is identical to the forward add round key transformation, because the XOR operation is its own inverse.

Example: ALZip



Example: ALZip 도움말

알집이 제공하는 암호화 알고리즘

알집에서 제공하는 암호화 알고리즘은 ZIP 2.0 호환, AES 128bit, AES 256bit, LEA128bit, LEA256bit 입니다. LEA 128bit, LEA 256bit 는 EGG 포맷에만 적용 가능합니다.

- ZIP 2.0 호환

  ZIP 에서 표준으로 사용하는 기본 암호화 알고리즘으로 수학적으로 검증되지는 않았지만 해독이 쉽 지 않으며, 가장 경량의 암호화 알고리즘입니다.

  개뻥

- AES 128bit block 사이즈가 128 key 길이는 3가지 있음

  Advanced Encryption Standard 의 약자로 현존하는 알고리즘 중 가장 널리 쓰이는 표준화된 알고리 즘이며, 128bit 크기의 블록 암호화를 사용한 대칭형 알고리즘입니다.

- AES 256bit block 사이즈는 128. key 256

  256bit 크기의 블록 암호화를 사용한 대칭형 알고리즘입니다.

Example: Adobe Acrobat