AES

목차

- DES is insecure
- AES 개요
- AES Data Units – State
- AES 구조 – Round
  - SubBytes
  - ShiftRows
  - MixColumns
  - AddRoundKey
- AES 구조 – Key Expansion
- AES 구조: Cipher vs. Inverse Cipher

DES is insecure

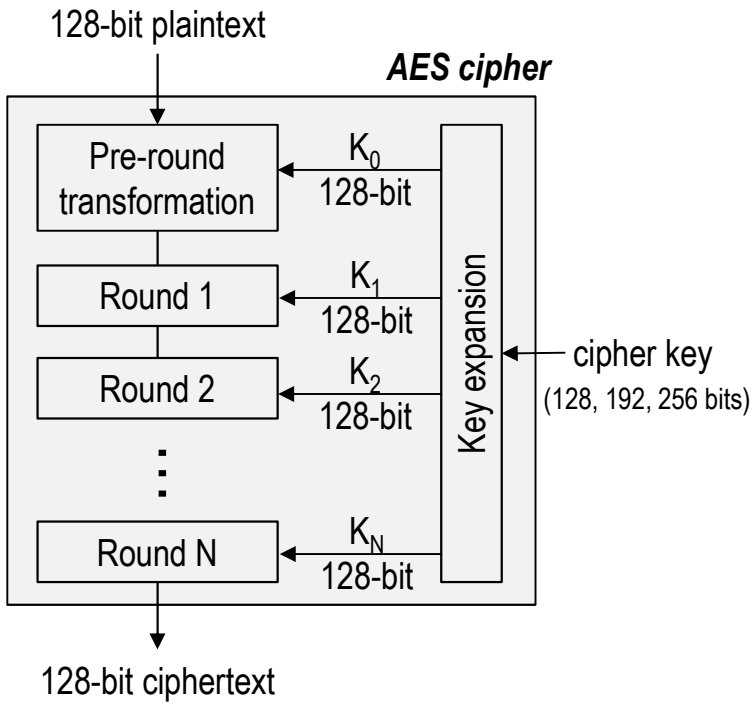
Reference: (Paar & Pelzl, 2010)

- DES 에 대한 주요 공격
  - DES 챌린지
    - RSA Security에 의해 시작
    - DES에 대한 brute force attack 대회
  - DES challenge I
    - 1997년 무작위 공격으로 4.5개월만에 해독
  - DES challenge II-1
    - 1998년 무작위 공격으로 39일만에 해독
  - DES challenge II-2
    - 1998년 무작위 공격으로 56시간만에 해독
  - DES challenge III
    - 1999년 무작위 공격으로 약 22시간만에 해독

AES

- AES (Advanced Encryption Standard)
  - History
    - 1997년, NIST는 DES를 대체할 AES라는 이름의 암호기술 공모
    - 1998년, 1st AES Candidate Conference 이후 21개 후보 중 15개 선정
    - 1999년, 2nd AES Candidate Conference 이후 15개 후보 중 5개 선정
      - MARS, RC6, Rijndael, Serpent, Twofish
    - 2000년, 3rd AES Candidate Conference 이후 Rijndael을 AES로 선정
    - 2001년, FIPS 197로 AES 발표
  - 선정기준
    - 안전성, 비용, 구현 효율성
  - Non-Feistel block cipher
    - 평문 블록 크기 128비트, 암호문 크기 128비트
    - 암호화 키 크기 128, 192, 256비트

AES 구조

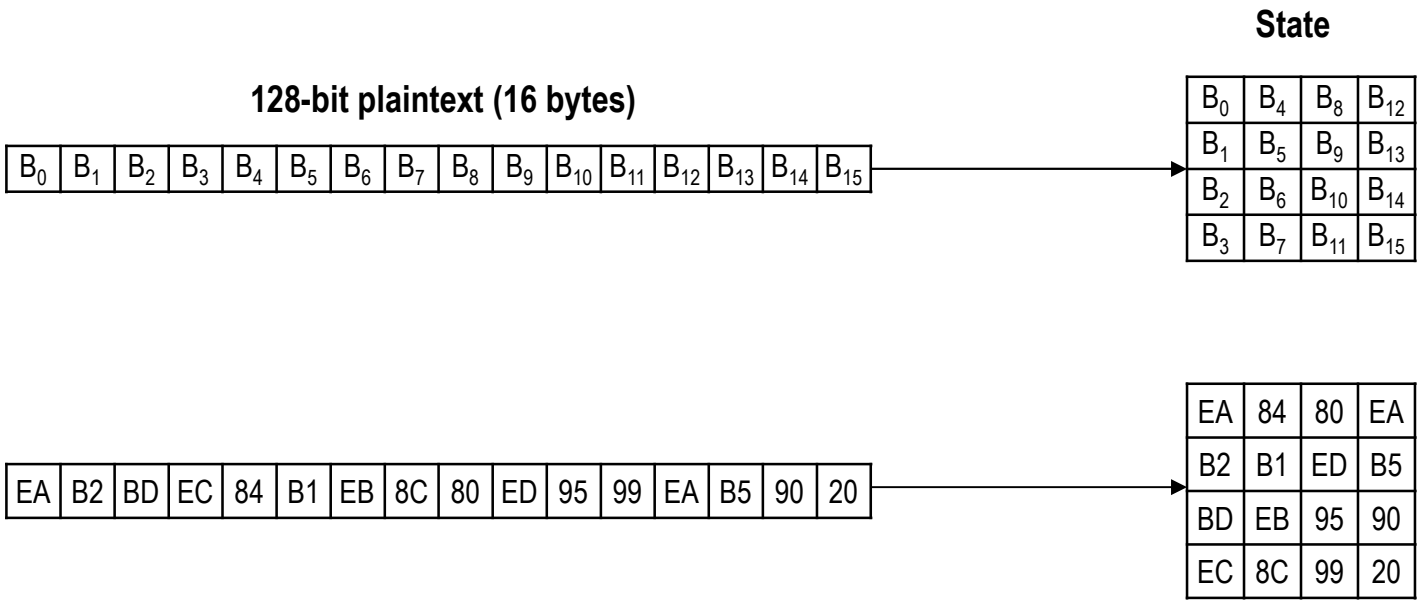


N	Key size
10	128
12	192
14	256

AES

- 평문 128비트, 암호문 128비트
- 키 크기 및 라운드 수에 따라 3가지 버전 존재: AES-128, AES-192, AES-256
- 암호화 키 크기가 다르더라도 key expansion 모듈이 만드는 round key는 항상 128비트임
- 복호화에서는 round 키 역순 적용
- Pre-round transformation 단계가 있어 라운드 키의 총 개수는 라운드 수보다 하나 많음

AES: Data Units – State

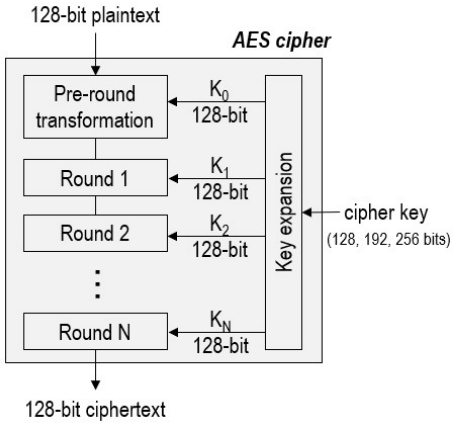


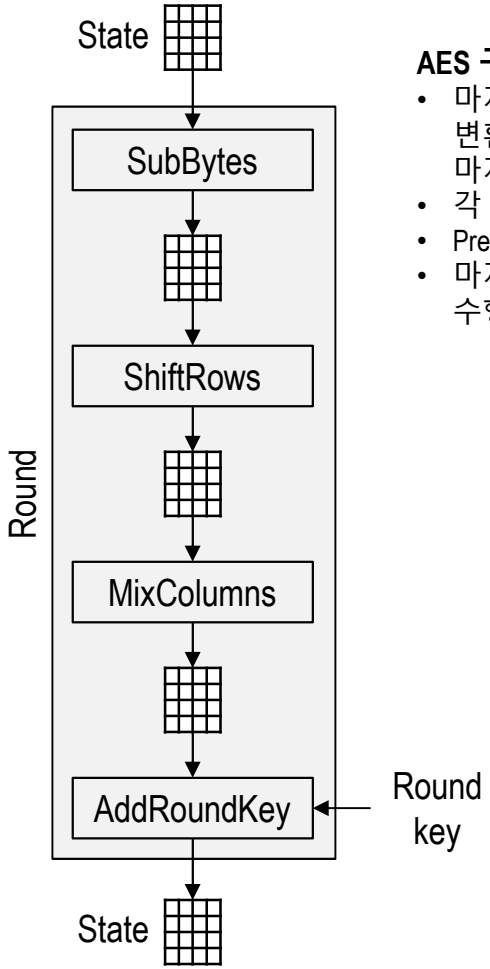
AES – State

- AES 알고리즘의 동작들은 내부적으로 State라는 (바이트들의) 2차원배열에 대해 적용됨
- 최초 128-bit 크기의 입력 블록은 State 배열에 복사된 후 내부 처리를 거침

AES 구조: Cipher의 Round

Reference: AES, FIPS PUB 197, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf



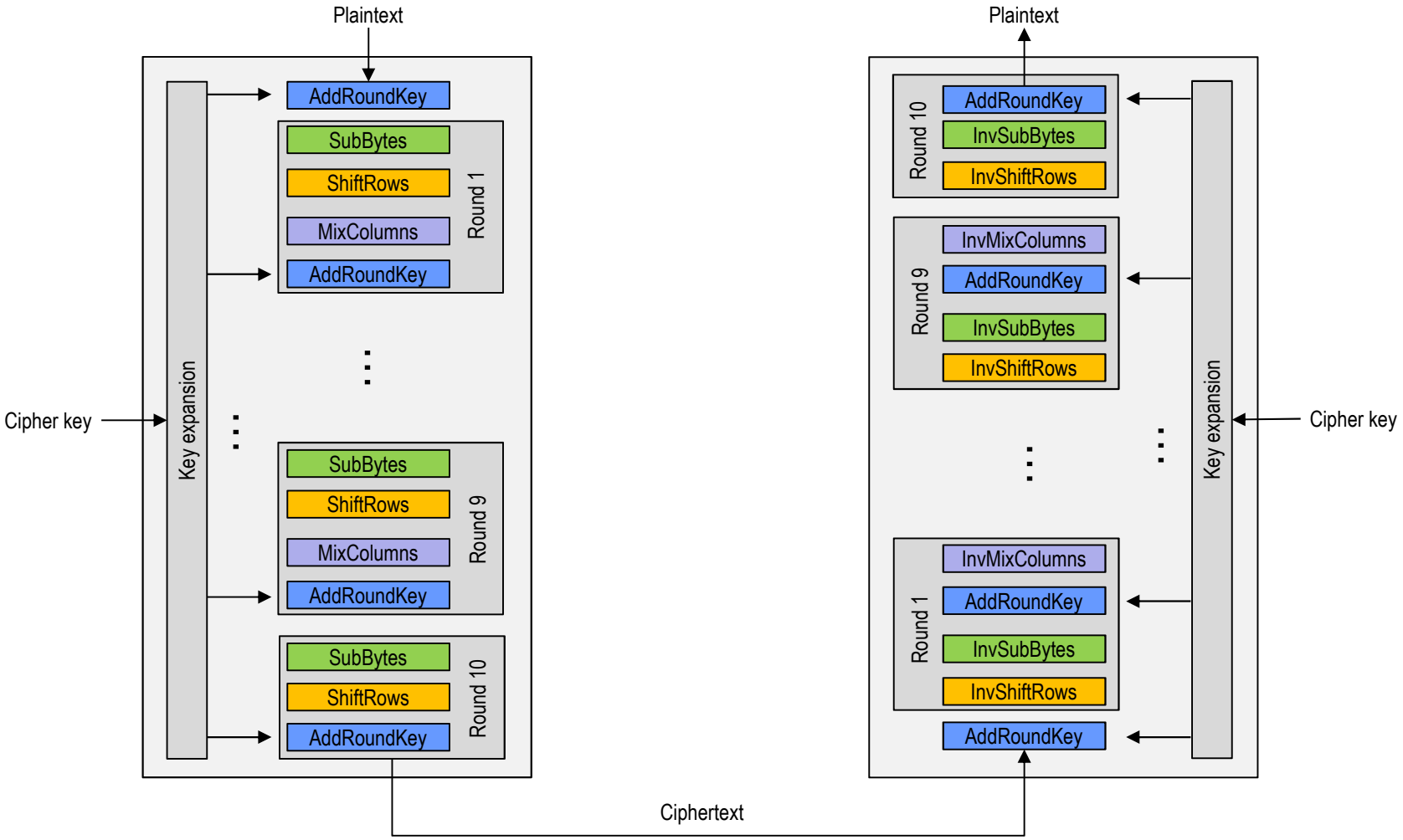


AES 구조: Cipher의 Round

- 마지막 라운드를 제외한 각 라운드는 가역적인 4개 변환(SubBytes, ShiftRows, MixColumns, AddRoundKey) 수행. 마지막 라운드는 3개 변환 수행
- 각 변환의 입출력은 state임
- Pre-round transformation은 하나의 변환 AddRoundKey만 수행
- 마지막 라운드는 MixColumns를 제외한 나머지 3개 변환 수행

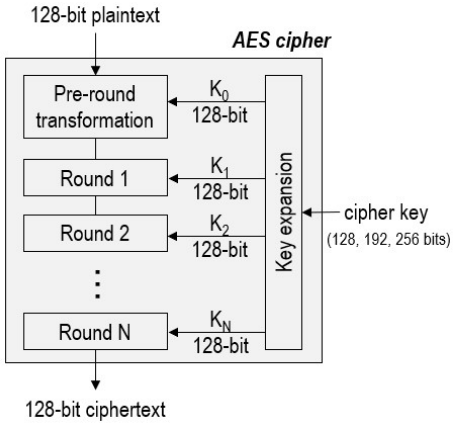
AES 구조: Cipher vs. Inverse Cipher – Design #1

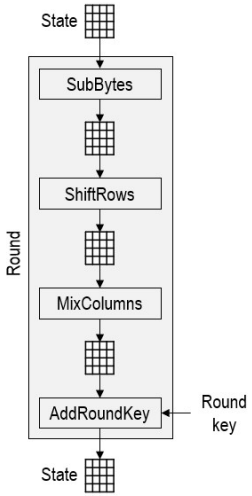
Reference: AES, FIPS PUB 197, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

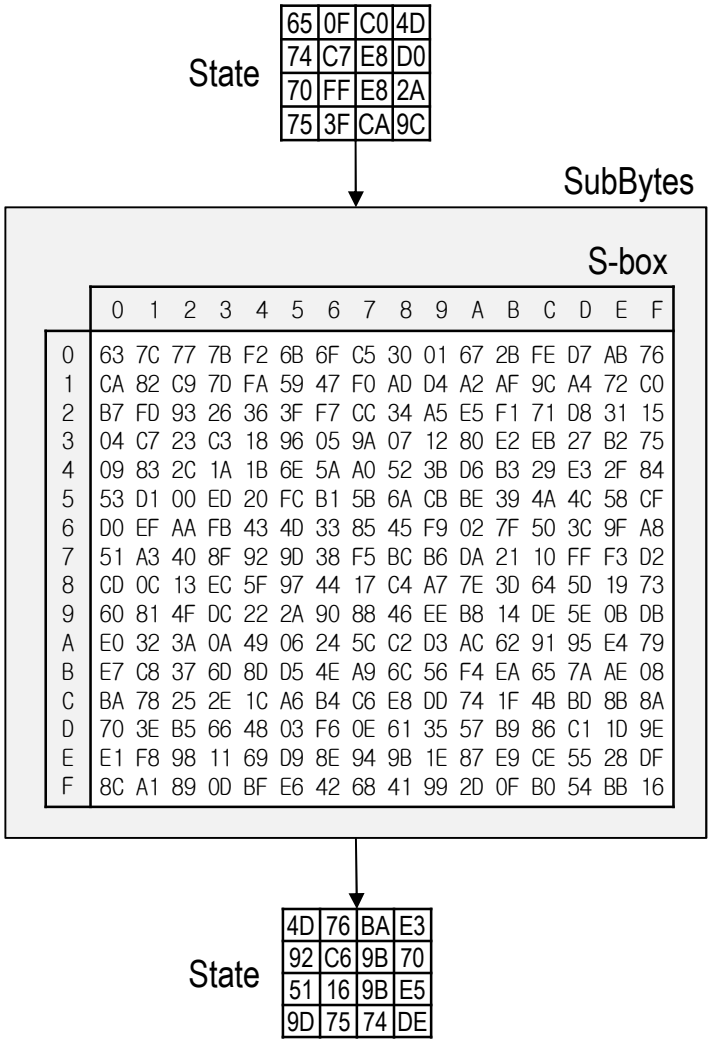


AES 구조: Round – SubBytes (테이블 기반)

Reference: AES, FIPS PUB 197, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf







	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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

AES 구조: Round – SubBytes

- SubBytes는 state를 입력 받아, state 내 각 바이트를 SubBytes S-box 테이블을 이용하여 새로운 바이트로 변환

AES 구조: Round – SubBytes, InvSubBytes

Reference: AES, FIPS PUB 197, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

State

65	0F	C0	4D
74	C7	E8	D0
70	FF	E8	2A
75	3F	CA	9C

SubBytes

S-box

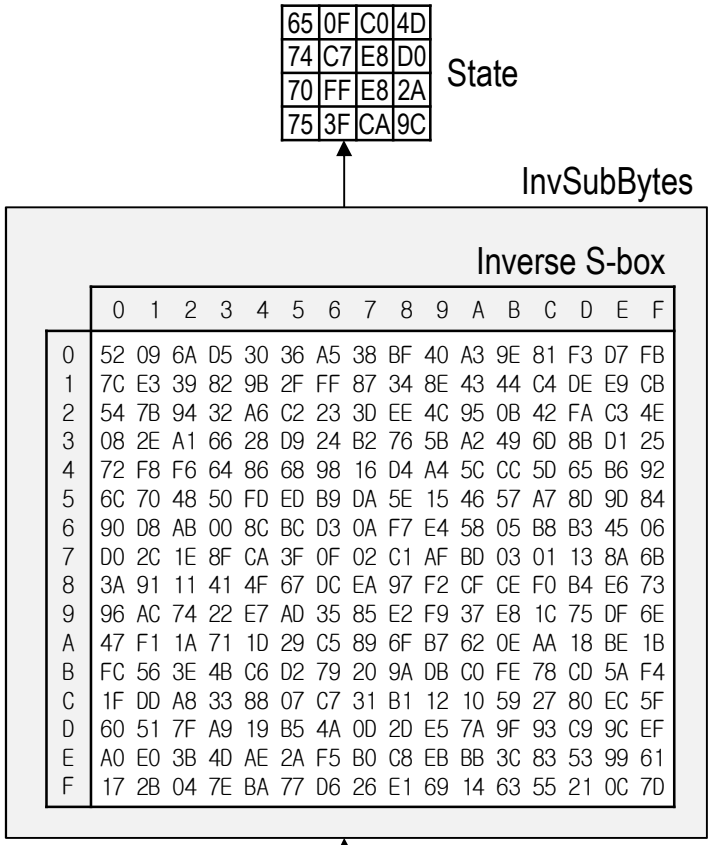
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	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

State

4D	76	BA	E3
92	C6	9B	70
51	16	9B	E5
9D	75	74	DE

AES 구조: Round – SubBytes, InvSubBytes

- 암호화에서는 SubBytes를 수행
- 복호화에서는 InvSubBytes를 수행
- SubBytes와 InvSubBytes는 서로의 역



AES 구조: Round – SubBytes (대수적 계산)

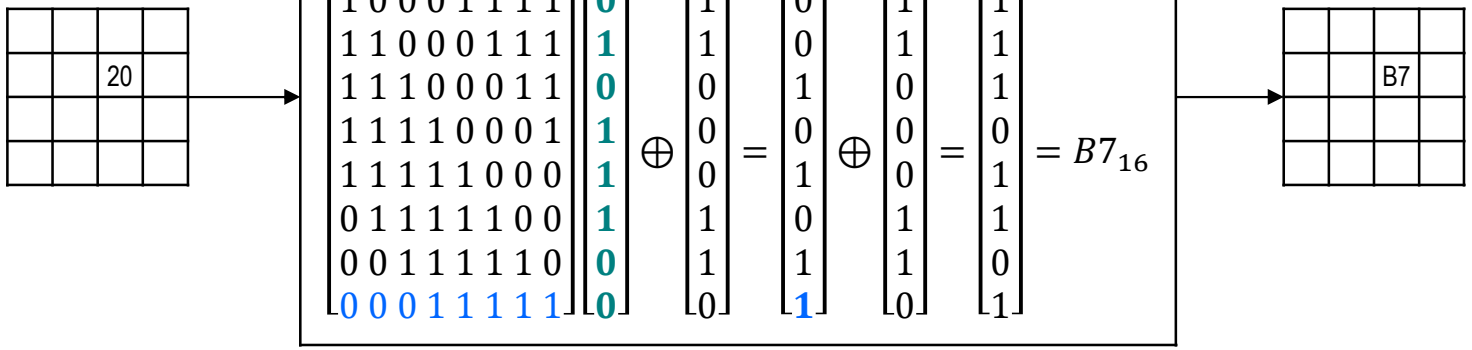
Reference: AES, FIPS PUB 197, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

q	r1	r2	r	t1	t2	t
x^3	$x^8+x^4+x^3+x+1$	x^5	x^4+x^3+x+1	0	1	x^3
$x+1$	x^5	x^4+x^3+x+1	x^3+x^2+1	1	x^3	x^4+x^3+1
x	x^4+x^3+x+1	x^3+x^2+1	1	x^3	x^4+x^3+1	$x^5+x^4+x^3+x$
x^3+x^2+1	x^3+x^2+1	1	0	x^4+x^3+1	$x^5+x^4+x^3+x$	0
	1	0		$x^5+x^4+x^3+x$	0	

Reference: (Forouzan, 2008, p.112)

SubBytes

20_{16}의 GF(2^8)에서의 곱셈의 역원은 $3A_{16}$
즉 $(x^5) \otimes (x^5+x^4+x^3+x)\ mod\ (x^8+x^4+x^3+x+1) = 1$



$$\begin{bmatrix}1&0&0&0&1&1&1&1\\1&1&0&0&0&1&1&1\\1&1&1&0&0&0&1&1\\1&1&1&1&0&0&0&1\\1&1&1&1&1&0&0&0\\0&1&1&1&1&1&0&0\\0&0&1&1&1&1&1&0\\0&0&0&1&1&1&1&1\end{bmatrix}\begin{bmatrix}\mathbf{0}\\\mathbf{1}\\\mathbf{0}\\\mathbf{1}\\\mathbf{1}\\\mathbf{1}\\\mathbf{0}\\\mathbf{0}\end{bmatrix} \oplus \begin{bmatrix}1\\1\\0\\0\\0\\1\\1\\0\end{bmatrix} = \begin{bmatrix}0\\0\\1\\0\\1\\0\\1\\\mathbf{1}\end{bmatrix} \oplus \begin{bmatrix}1\\1\\0\\0\\0\\1\\1\\0\end{bmatrix} = \begin{bmatrix}1\\1\\0\\1\\1\\0\\0\\1\end{bmatrix} = B7_{16}$$

$3A_{16} = \mathbf{00111010}$

$0\cdot 0 \oplus 0\cdot 1 \oplus 0\cdot 0 \oplus \mathbf{1}\cdot\mathbf{1} \oplus \mathbf{1}\cdot\mathbf{1} \oplus \mathbf{1}\cdot\mathbf{1} \oplus 1\cdot 0 \oplus 1\cdot 0$
$= 1 \oplus 1 \oplus 1$
$= \mathbf{1}$

다항식(polynomial)의 계수 연산 → GF(2)에서 수행

AES 구조: Round – SubBytes (대수적 계산)

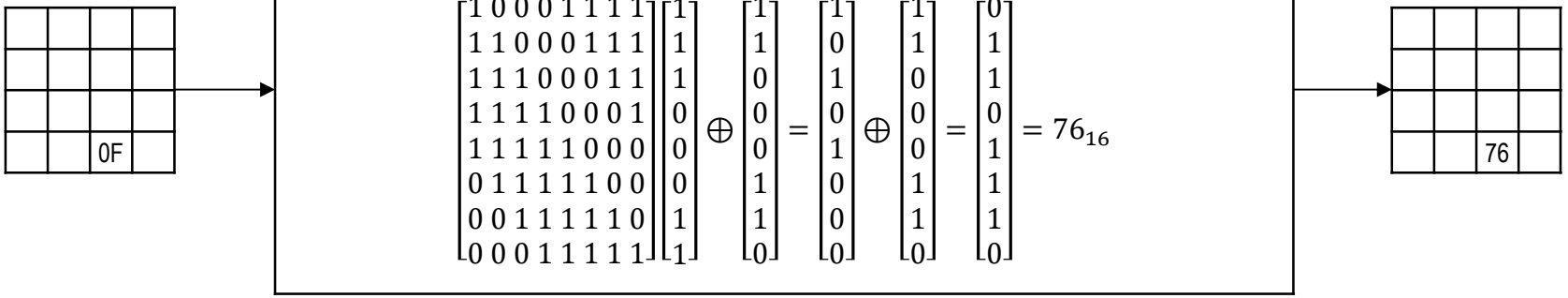
Reference: AES, FIPS PUB 197, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

q	r1	r2	r	t1	t2	t
x^5+x^4+1	$x^8+x^4+x^3+x+1$	x^3+x^2+x+1	x^2	0	1	x^5+x^4+1
$x+1$	x^3+x^2+x+1	x^2	$x+1$	1	x^5+x^4+1	x^6+x^4+x
$x+1$	x^2	$x+1$	1	x^5+x^4+1	x^6+x^4+x	$x^7+x^6+x^2+x+1$
$x+1$	$x+1$	1	0	x^6+x^4+x	$x^7+x^6+x^2+x+1$	0
	1	0		$x^7+x^6+x^2+x+1$	0	

SubBytes

$0F_{16}$의 GF(2^8)에서의 곱셈의 역원은 $C7_{16}$

즉 $(x^3+x^2+x+1) \otimes (x^7+x^6+x^2+x+1)\ mod\ (x^8+x^4+x^3+x+1) = 1$



$$\begin{bmatrix}1&0&0&0&1&1&1&1\\1&1&0&0&0&1&1&1\\1&1&1&0&0&0&1&1\\1&1&1&1&0&0&0&1\\1&1&1&1&1&0&0&0\\0&1&1&1&1&1&0&0\\0&0&1&1&1&1&1&0\\0&0&0&1&1&1&1&1\end{bmatrix}\begin{bmatrix}1\\1\\1\\0\\0\\0\\1\\1\end{bmatrix} \oplus \begin{bmatrix}1\\1\\0\\0\\0\\1\\1\\0\end{bmatrix} = \begin{bmatrix}1\\0\\1\\0\\1\\0\\0\\0\end{bmatrix} \oplus \begin{bmatrix}1\\1\\0\\0\\0\\1\\1\\0\end{bmatrix} = \begin{bmatrix}0\\1\\1\\0\\1\\1\\1\\0\end{bmatrix} = 76_{16}$$

AES 구조: Round – SubBytes (대수적 계산)

Reference: AES, FIPS PUB 197, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

State

65	0F	C0	4D
74	C7	E8	D0
70	FF	E8	2A
75	3F	CA	9C

SubBytes

아래 절차를 State 내 각 바이트에 대해 반복 수행

1. 입력 State 내 r행 c열의 바이트 $s_{r,c}$에 대해 GF(2^8)에서의 곱셈의 역원 $\boldsymbol{b}$를 계산한 후, 미리 정해진 행렬 $\boldsymbol{X}$와 벡터 $\boldsymbol{c}$에 대해 $\boldsymbol{v} = \boldsymbol{X}\boldsymbol{b} + \boldsymbol{c}$를 계산

$$\boldsymbol{v} \quad \boldsymbol{X} \quad \boldsymbol{b} \quad \boldsymbol{c}$$

$$\begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix} = \begin{bmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

2. 앞에서 계산된 $\boldsymbol{v}$를 출력 State 내 r행 c열의 바이트 $s_{r,c}$에 저장

State

4D	76	BA	E3
92	C6	9B	70
51	16	9B	E5
9D	75	74	DE

$$\begin{bmatrix} 0&0&1&0&0&1&0&1 \\ 1&0&0&1&0&0&1&0 \\ 0&1&0&0&1&0&0&1 \\ 1&0&1&0&0&1&0&0 \\ 0&1&0&1&0&0&1&0 \\ 0&0&1&0&1&0&0&1 \\ 1&0&0&1&0&1&0&0 \\ 0&1&0&0&1&0&1&0 \end{bmatrix} \begin{bmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{bmatrix} = \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix}$$

다항식(polynomial)의 계수 연산은 GF(2)에서 수행

State

65	0F	C0	4D
74	C7	E8	D0
70	FF	E8	2A
75	3F	CA	9C

InvSubBytes

아래 절차를 State 내 각 바이트에 대해 반복 수행

1. 입력 State 내 r행 c열의 바이트 v에 대해, 미리 정해진 행렬 $\boldsymbol{Y}$와 벡터 $\boldsymbol{c}$에 대해 $\boldsymbol{b} = \boldsymbol{Y}(\boldsymbol{v} - \boldsymbol{c})$ 를 계산

$$\boldsymbol{b} \quad \boldsymbol{Y} \quad \boldsymbol{v} \quad \boldsymbol{c}$$

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \begin{bmatrix} 0&0&1&0&0&1&0&1 \\ 1&0&0&1&0&0&1&0 \\ 0&1&0&0&1&0&0&1 \\ 1&0&1&0&0&1&0&0 \\ 0&1&0&1&0&0&1&0 \\ 0&0&1&0&1&0&0&1 \\ 1&0&0&1&0&1&0&0 \\ 0&1&0&0&1&0&1&0 \end{bmatrix} \left(\begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \right)$$

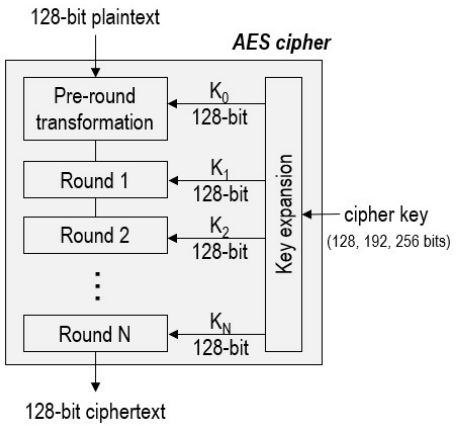
2. 앞에서 계산된 $\boldsymbol{b}$의 GF(2^8)에서의 곱셈의 역원을 계산하여 출력 State 내 r행 c열의 바이트 $s_{r,c}$에 저장

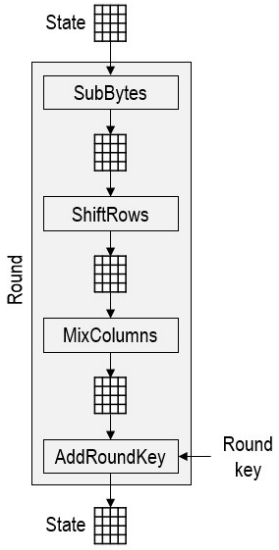
State

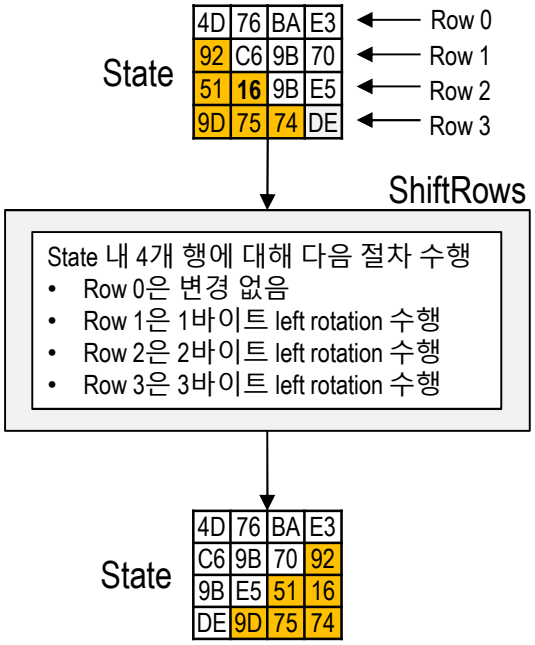
4D	76	BA	E3
92	C6	9B	70
51	16	9B	E5
9D	75	74	DE

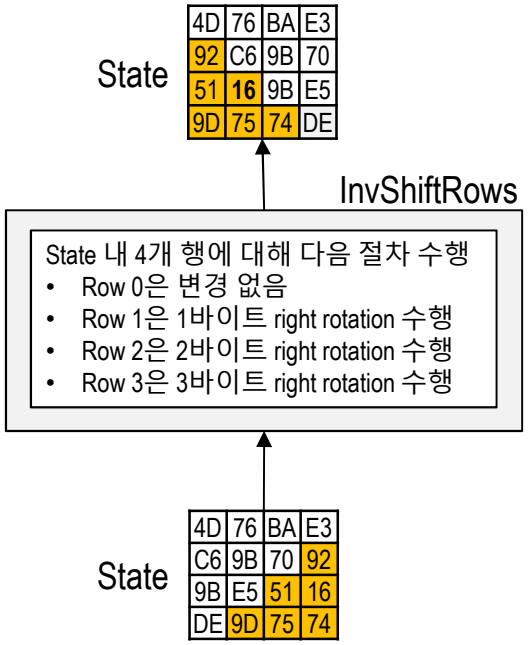
AES 구조: Round – ShiftRows, InvShiftRows

Reference: AES, FIPS PUB 197, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf







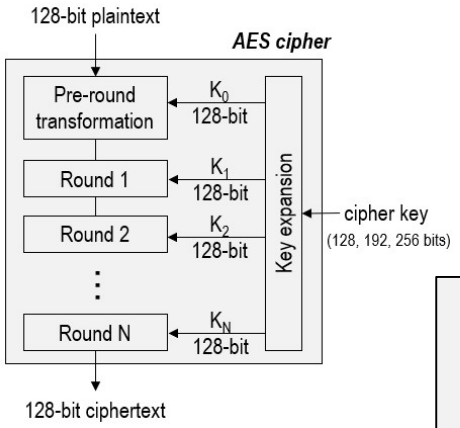


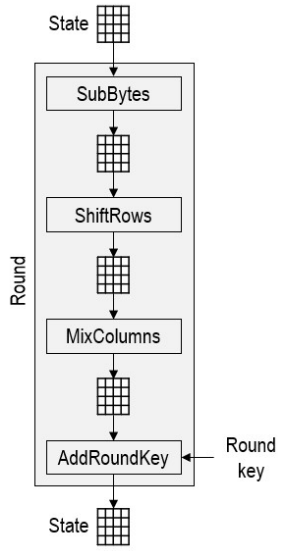
AES 구조: Round – ShiftRows, InvShiftRows

- 암호화에서는 ShiftRows를 수행
- 복호화에서는 InvShiftRows를 수행
- ShiftRows와 InvShiftRows는 서로의 역
- 바이트 단위로 permutation을 수행하는 것이며 특정 바이트 내 비트 순서는 변경 없음
- DES의 경우 비트 단위로 permutation 수행되었음

AES 구조: Round – MixColumns

Reference: AES, FIPS PUB 197, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf





State

4D	76	BA	E3
C6	9B	70	92
9B	E5	51	16
DE	9D	75	74

MixColumns

아래 절차를 State 내 각 열에 대해 반복 수행

1. 입력 State 내 i 열의 바이트들 $s_{0,i}, s_{1,i}, s_{2,i}, s_{3,i}$ 에 대해 다음 수식을 통해 i 열의 새로운 바이트들 $s'_{i,0}, s'_{i,1}, s'_{i,2}, s'_{i,3}$을 계산한다

$$\begin{bmatrix} s'_{i,0} \\ s'_{i,1} \\ s'_{i,2} \\ s'_{i,3} \end{bmatrix} = \begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} \begin{bmatrix} s_{0,i} \\ s_{1,i} \\ s_{2,i} \\ s_{3,i} \end{bmatrix}$$

2. 앞에서 계산된 i 열의 새로운 바이트들 $s'_{i,0}, s'_{i,1}, s'_{i,2}, s'_{i,3}$을 출력 State 내 i 열의 바이트들에 저장

State

8E	22	DB	12
B2	F2	DC	92
DF	80	F7	C1
2D	C5	1E	52

4D	**76**	BA	E3
C6	**9B**	70	92
9B	**E5**	51	16
DE	**9D**	75	74

$$\begin{bmatrix} 22 \\ F2 \\ 80 \\ C5 \end{bmatrix} = \begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} \begin{bmatrix} 76 \\ 9B \\ E5 \\ 9D \end{bmatrix}$$

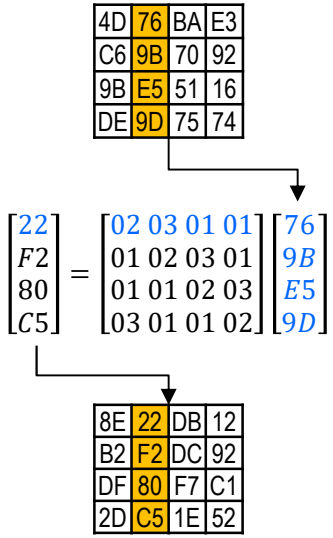
8E	**22**	DB	12
B2	**F2**	DC	92
DF	**80**	F7	C1
2D	**C5**	1E	52

AES 구조: Round – MixColumns

- SubBytes는 state 내 바이트 변경 시 해당 바이트 값에만 의존
- MixColumns의 경우 state 내 바이트 변경 시 해당 바이트가 속해 있는 열(column) 내 다른 바이트들의 값에 의존

AES 구조: Round – MixColumns

Reference: AES, FIPS PUB 197, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf



$$(\{02\}\cdot\{76\})\oplus(\{03\}\cdot\{9B\})\oplus(\{01\}\cdot\{E5\})\oplus(\{01\}\cdot\{9D\})=\{22\}$$

GF(2^8)에서의 다항식(polynomial) 연산

$\{02\}=00000010=x$
$\{76\}=01110110=x^6+x^5+x^4+x^2+x$
$\{02\}\cdot\{76\}=\big((x)\cdot(x^6+x^5+x^4+x^2+x)\big)\ mod\ (x^8+x^4+x^3+x+1)=x^7+x^6+x^5+x^3+x^2$
$=11101100$

$\{03\}=00000011=x+1$
$\{9B\}=10011011=x^7+x^4+x^3+x+1$
$\{03\}\cdot\{9B\}=\big((x+1)\cdot(x^7+x^4+x^3+x+1)\big)\ mod\ (x^8+x^4+x^3+x+1)=x^7+x^5+x^4+x^2+x$
$=10110110$

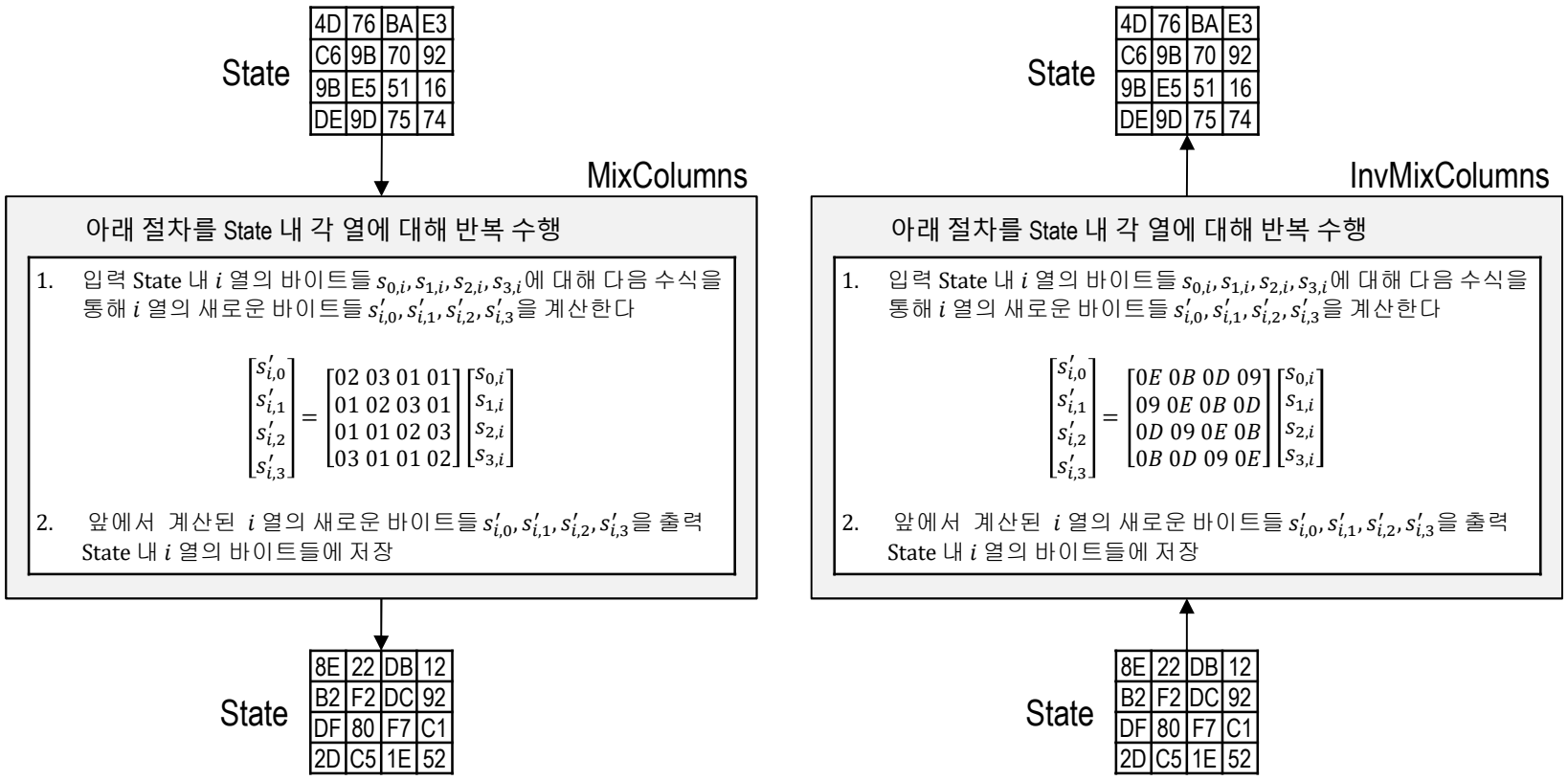
$\{01\}\cdot\{E5\}=x^7+x^6+x^5+x^2+1$
$=11100101$

$\{01\}\cdot\{9D\}=x^7+x^4+x^3+x^2+1$
$=10011101$

$(\{02\}\cdot\{76\})\oplus(\{03\}\cdot\{9B\})\oplus(\{01\}\cdot\{E5\})\oplus(\{01\}\cdot\{9D\})$
$=11101100\oplus10110110\oplus11100101\oplus10011101=00100010=\{22\}$

AES 구조: Round – MixColumns, InvMixColumns

Reference: AES, FIPS PUB 197, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf



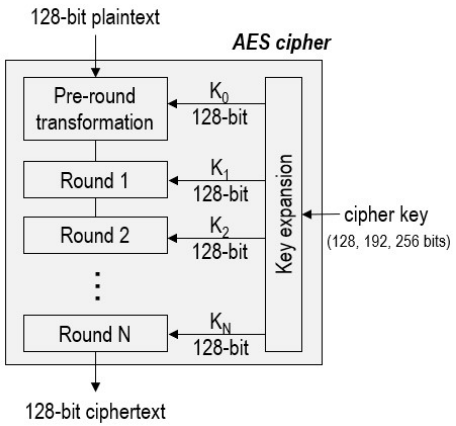
AES 구조: Round – MixColumns, InvMixColumns

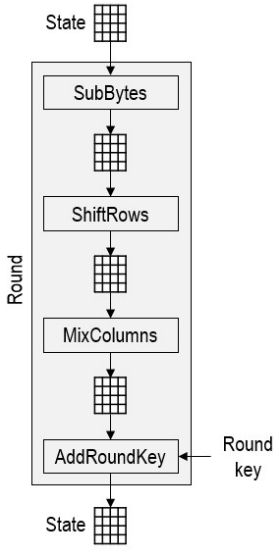
- 암호화에서는 MixColumns를 수행
- 복호화에서는 InvMixColumns를 수행
- MixColumns와 InvMixColumns는 서로의 역

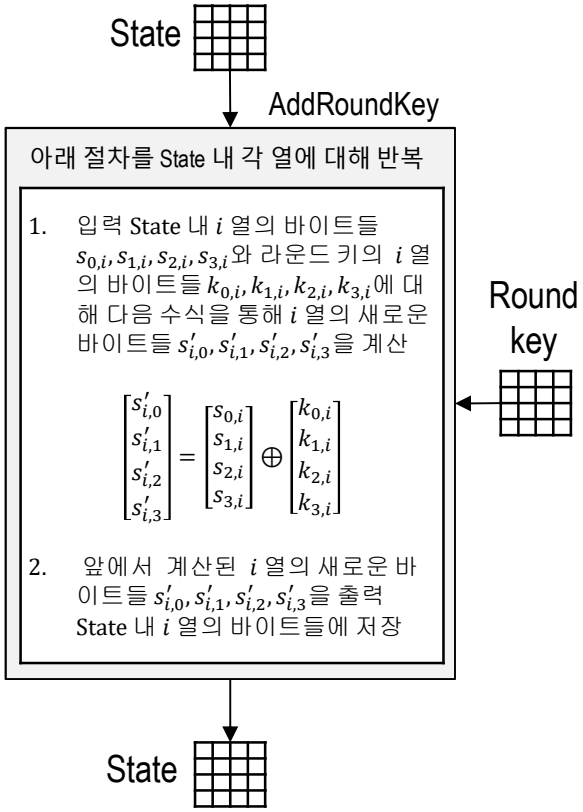
$$\begin{bmatrix} 0E\ 0B\ 0D\ 09 \\ 09\ 0E\ 0B\ 0D \\ 0D\ 09\ 0E\ 0B \\ 0B\ 0D\ 09\ 0E \end{bmatrix} \begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} = \begin{bmatrix} 01\ 00\ 00\ 00 \\ 00\ 01\ 00\ 00 \\ 00\ 00\ 01\ 00 \\ 00\ 00\ 00\ 01 \end{bmatrix}$$

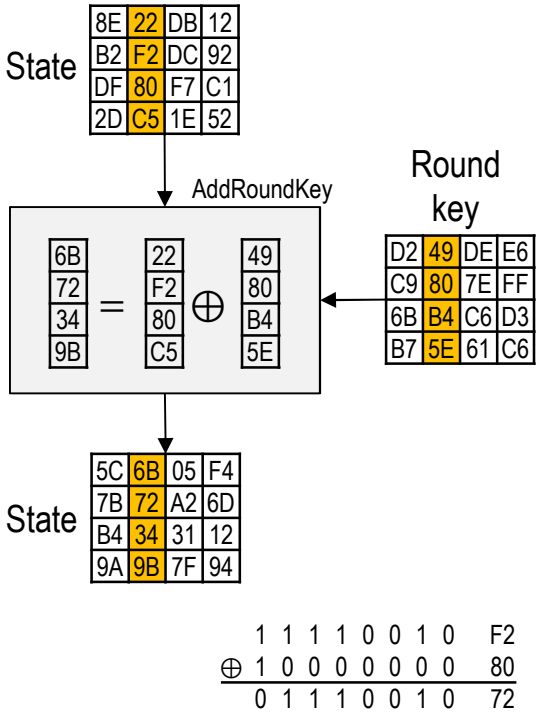
AES 구조: Round – AddRoundKey

Reference: AES, FIPS PUB 197, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf









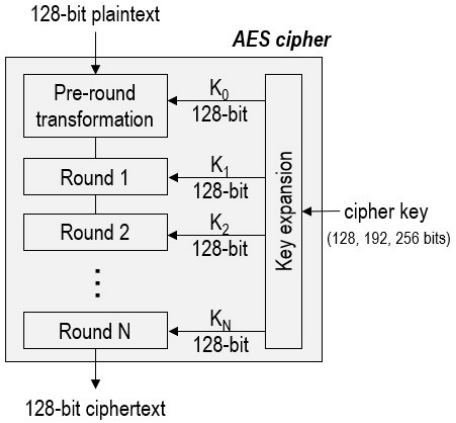
$$\begin{array}{rcccccccc|c} & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & F2 \\ \oplus & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 80 \\ \hline & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 72 \end{array}$$

AES 구조: Round – AddRoundKey

- AES는 최초 암호화 키로부터 key expansio을 통해 pre-round transformation 단계 및 N개 각 라운드에 사용될 128비트 크기의 라운드 키가 N+1개 생성됨
- AddRoundKey는 128비트 state와 라운드 키 128비트를 bitwise XOR하여 새로운 state를 생성한다
- AddRoundKey는 self-invertible

AES 구조: Key Expansion

Reference: AES, FIPS PUB 197, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf



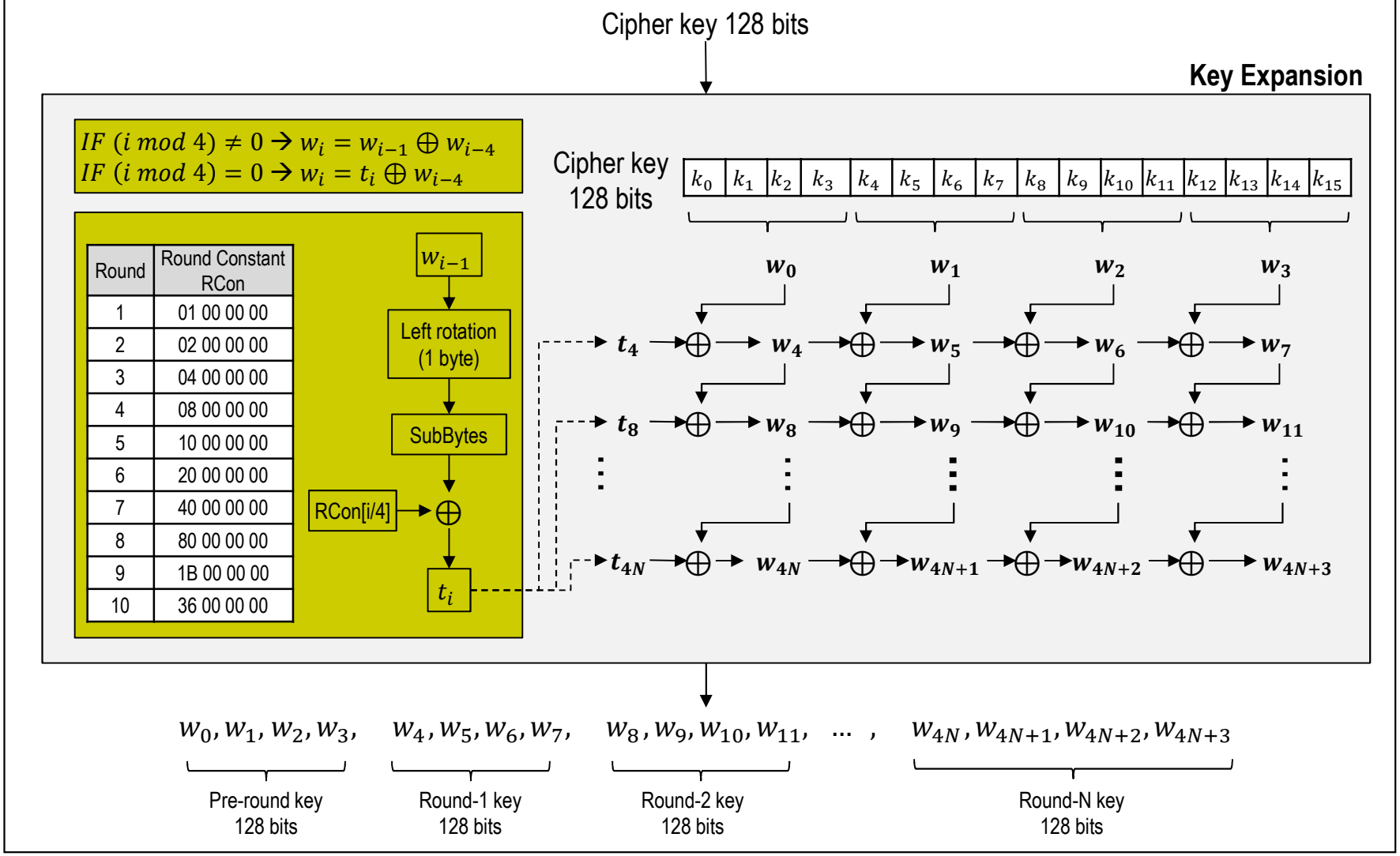
AES 구조: Key Expansion

- 라운드 수가 N인 경우, 최초 하나의 암호화 키로부터, $N+1$ 개 128-bit 라운드 키 생성
- 워드(word) → AES에서 4 바이트를 지칭하는 용어
- 라운드 키는 워드 단위로 생성됨
- 라운드 수가 N인 경우, key expansion은 다음과 같이 $4(N+1)$개의 워드들을 생성
  → $w_0, w_1, w_2, \ldots, w_{4(N+1)-1}$
- AES-128의 경우 10 라운드이므로 총 44개 워드 생성
- AES-192의 경우 12 라운드이므로 총 52개 워드 생성
- AES-128의 경우 14 라운드이므로 총 60개 워드 생성
- 각 라운드 키는 4개 워드들로 구성됨

Round	Words
Pre-round	w_0, w_1, w_2, w_3
1	w_4, w_5, w_6, w_7
2	w_8, w_9, w_{10}, w_{11}
…	…
N	$w_{4N}, w_{4N+1}, w_{4N+2}, w_{4N+3}$

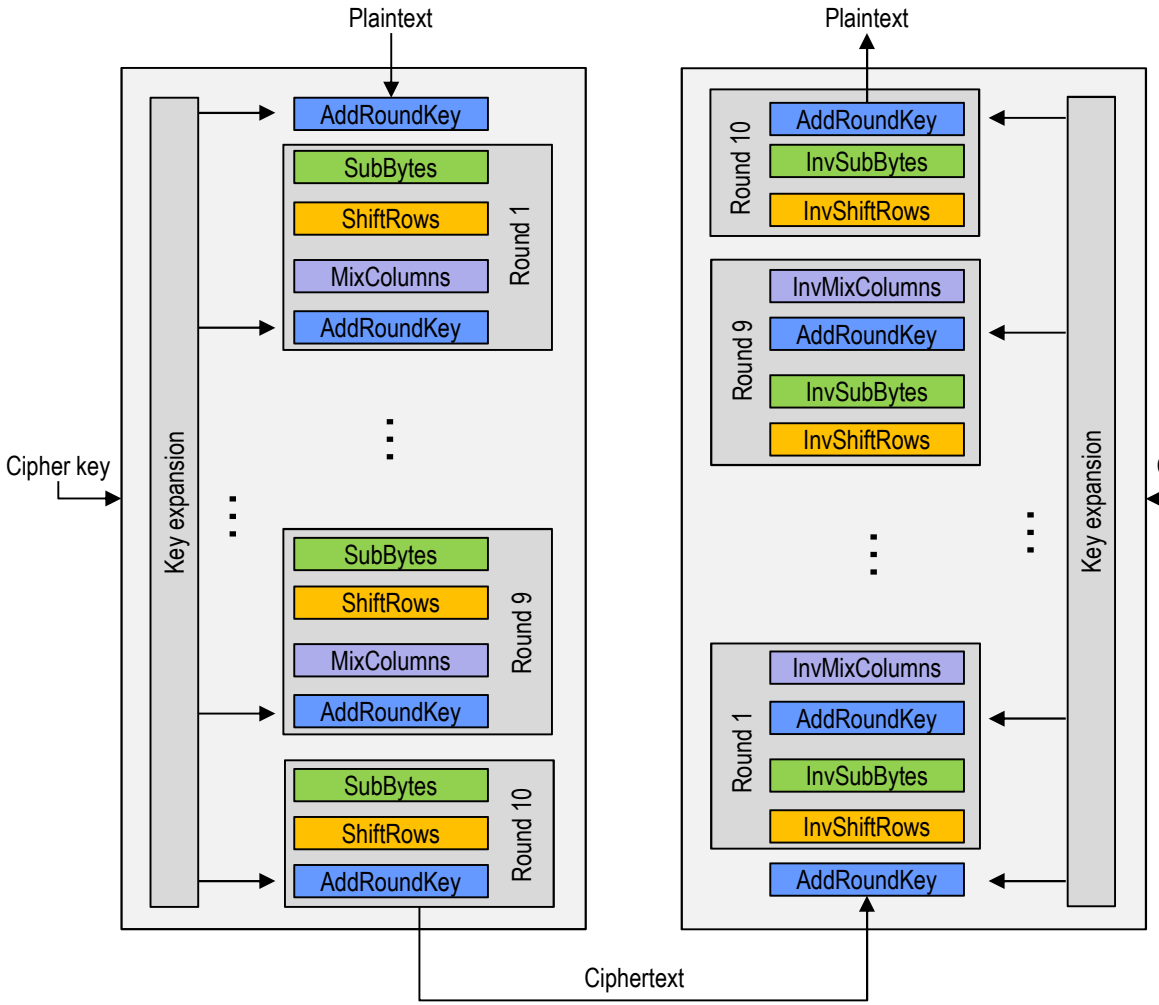
AES 구조: Key Expansion

Reference: AES, FIPS PUB 197, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf



AES 구조: Cipher vs. Inverse Cipher – Design #1

Reference: AES, FIPS PUB 197, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf



Cipher vs. Inverse cipher

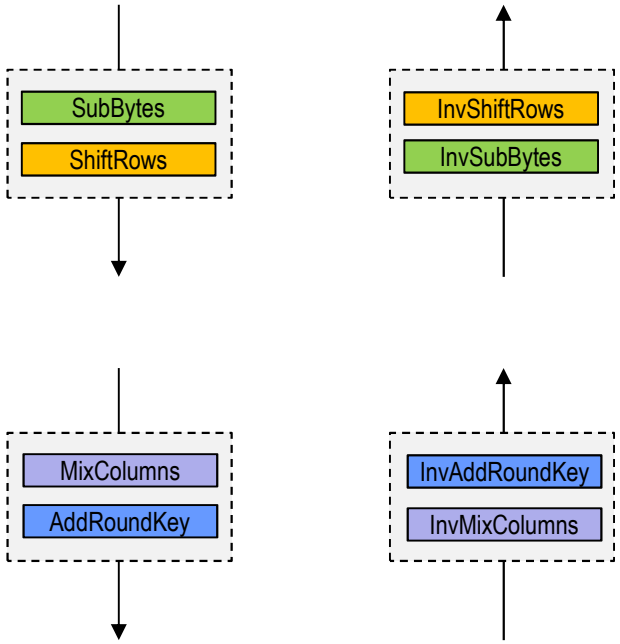
- AES는 non-Feistel cipher이므로 각 변환은 가역적임. 즉 cipher와 inverse cipher의 대응하는 변환은 서로의 역
- 단 라운드 키들은 역순 적용 필요
- 그러나 Design #1은 각 라운드의 변환 순서가 cipher와 inverse cipher에서 동일하지 않음. SubBytes와 ShiftRows의 적용 순서가 반대이며, MixColumns와 AddRoundKey의 적용 순서 역시 반대임

Design #1의 단점

- 암호화 모듈과 복호화 모듈의 구조가 동일하지 않음
- 암복호화를 수행하는 장치에서 분리된 암호화 모듈과 복호화 모듈이 필요

AES 구조: Inverse Cipher – 설계 변경 착안점

Reference: AES, FIPS PUB 197, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf



설계 변경 목표
- 암호화 모듈과 복호화 모듈이 동일한 구조를 갖도록 함
- 즉 암호화 모듈과 복호화 모듈의 변환 순서가 동일하도록 함

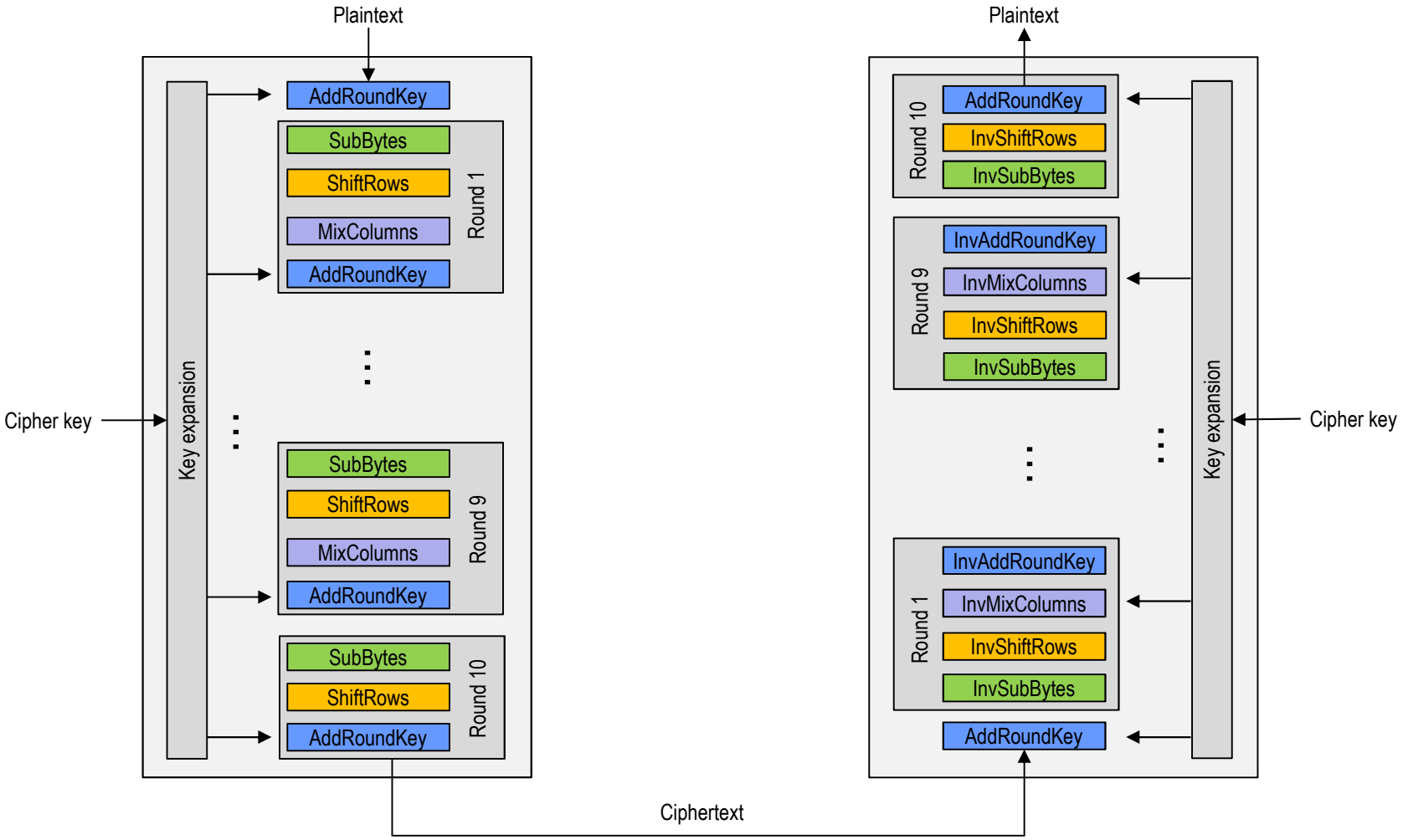
설계 변경 착안점 1
- SubBytes는 바이트의 내용을 변경하고, ShiftRows는 바이트의 순서를 변경함
- 즉 두 변환의 순서 변경은 SubBytes와 ShiftRows 변환 그룹 측면의 가역성에 영향 없음
- Inverse cipher에서 InvSubBytes와 InvShiftRows의 적용 순서를 변경함

설계 변경 착안점 2
- AddRoundKey에서는 라운드 키의 각 열을 state의 각 열과 XOR 수행하나, InvAddRoundKey에서는 라운드 키의 각 열에 InvMixColumns를 적용한 결과를 state의 각 열과 XOR을 수행함
- AddRoundKey 대신 InvAddRoundKey를 사용하면, InvMixColumns와 InvAddRoundKey의 순서를 변경하더라도 MixColumns과 AddRoundKey 변환 그룹 측면의 가역성에 영향 없음
- InvMixColumns와 InvAddRoundKey의 적용 순서를 변경함

State 행렬 $\boldsymbol{S}$, MixColumns에서의 상수 행렬 $\boldsymbol{X}$, InvMixColumns에서의 상수 행렬 $\boldsymbol{Y}$, AddRoundKey에서의 라운드 키 $\boldsymbol{K}$에 대해,
암호화 모듈의 MixColumns와 AddRoundKey 변환들의 출력은 $\boldsymbol{XS} \oplus \boldsymbol{K}$
복호화 모듈의 InvMixColumns와 InvAddRoundKey 변환들의 출력은 $\big(\boldsymbol{Y}(\boldsymbol{XS} \oplus \boldsymbol{K})\big) \oplus \boldsymbol{YK} = \boldsymbol{YXS} \oplus \boldsymbol{YK} \oplus \boldsymbol{YK} = \boldsymbol{YXS} = \boldsymbol{S}$

AES 구조: Cipher vs. Inverse Cipher – Design #2

Reference: AES, FIPS PUB 197, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf



References

- Behrouz A. Forouzan, Cryptography and Network Security, McGraw-Hill, 2008
- William Stallings, Cryptography and Network Security: Principles and Practice, Sixth Edition, Prentice Hall, 2014
- Christof Paar, Jan Pelzl, Understanding Cryptography: A Textbook for Students and Practitioners, Springer, 2010
- 김명환, 수리암호학개론, 2019
- 정민석, 암호수학, 경문사, 2017
- 최은미, 정수와 암호론, 북스힐, 2019
- 이민섭, 정수론과 암호론, 교우사, 2008