AES
(Advanced Encryption Standard)

https://youtu.be/mlH3-Y4sPdk

HSU 한성대학교 HANSUNG UNIVERSITY

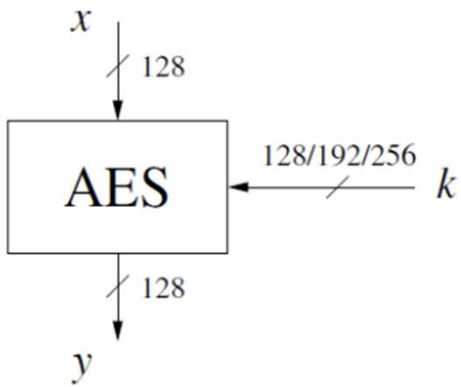
CryptoCraft LAB

AES 알고리즘 개요

갈루아 체(Galois Fields) 개요

AES의 내부 구조

AES 알고리즘 개요



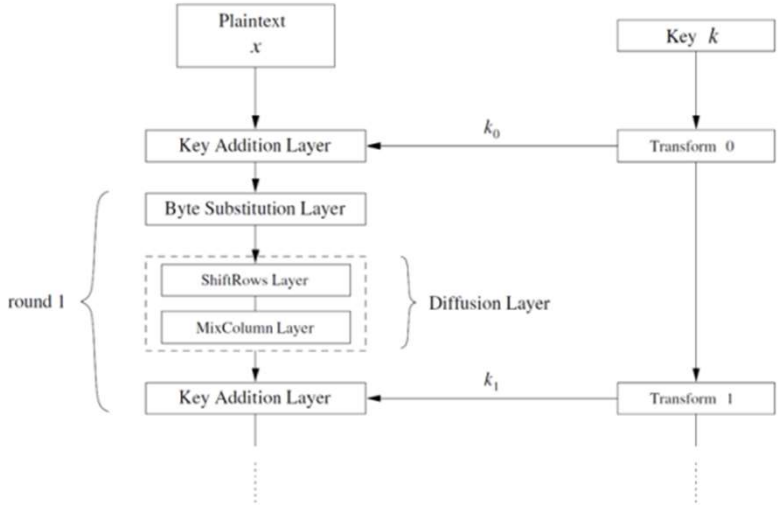
AES에서의 입출력의 블록 크기는 128bit로 고정되어 있으나,
키 길이(Key Length)는 128, 192, 256 bit가 가능하며,
그에 따라 라운드의 수도 달라짐

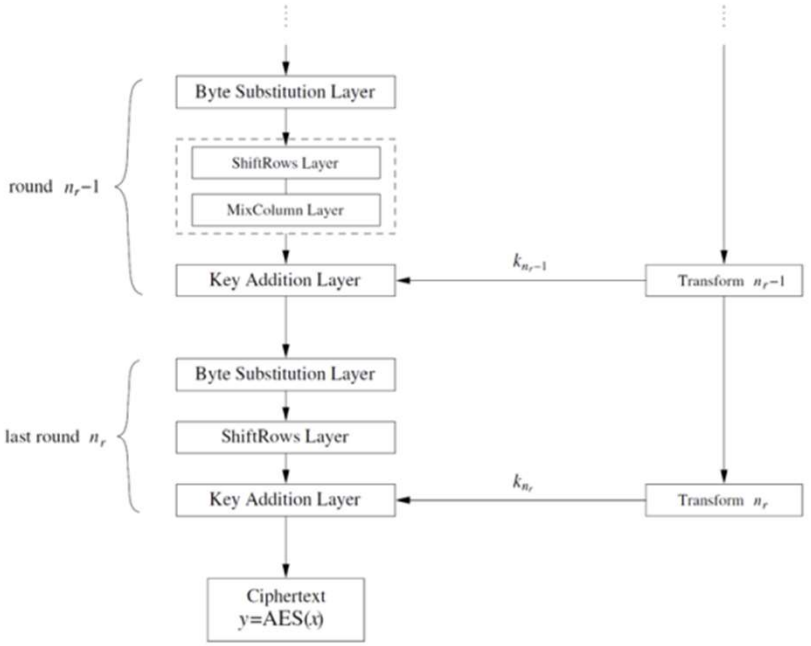
key lengths	# rounds = n_r
128 bit	10
192 bit	12
256 bit	14

AES 알고리즘 개요

10/12 /14 라운드를 갖는 반복적인 구조임

각 라운드는 여러 계층(Layers)으로 구성됨.





AES 알고리즘 개요

각 계층은 데이터 경로(Data Path)의 128bit 전체를 다루며, 각 데이터 경로를 알고리즘의 상태(State)라고 함.

키 덧셈 계층(Key Addition Layer)

키 스케줄에서 메인 키(Main-Key)에서 파생된 128 bit 라운드 키(Round Key) 또는 서브 키(Sub-Key)가 상태(State)에 XOR됨.

바이트 대치 계층(Byte Substitution Layer)(S-box)

각 상태의 요소는 특별한 수학적 특성을 가지는 Lookup Table을 이용하여 비선형적으로 변형되어 데이터에 혼돈(Confusion)을 가함. 즉, 각각의 상태 비트의 변화가 데이터 경로를 통해 빠르게 전달됨.

확산 계층(Diffusion Layer)

모든 상태 비트에 확산을 제공함.

ShiftRows Layer – 바이트 레벨로 데이터를 전치(Permution)함.

MixColumn Layer – 4 byte 블록을 결합하는 행렬 연산임.

갈루아 체 개론

AES의 대부분의 계층, 특히 S-Box 및 MixColumn 계층에서 갈루아 체 연산이 이용됨.

유한체(Finite fields)의 존재

갈루아 체라고도 불리는 유한체는 한정된 원소들의 집합임.

대체로 유한체 내에서 원소들끼리의 덧셈, 뺄셈, 곱셉 등이 가능함.

Definition 4.3.1 군(Group): 집합 G와 연산 $\circ$로 구성됨(i.e., $(G,\circ)$)

- 군 연산(Group Operation) $\circ$는 닫혀 있음. 즉, for all $a, b \in G$, $a \circ b = c \in G$.
- 군 연산은 결합법칙(Associative)이 성립함. 즉, for all $a, b, c \in G$, $a \circ (b \circ c) = (a \circ b) \circ c$.
- 항등원(Neutral Element or Identity Element) $e \in G$이 존재함. 즉, for all $a \in G$, $a \circ e = e \circ a = a$.
- 각 $a \in G$에 대해 $a \circ a^{-1} = a^{-1} \circ a = e$인 a의 역원(Inverse) $a^{-1} \in G$이 존재함.
- 모든 군 $a, b \in G$에 대해 $a \circ b = b \circ a$이 성립하면, 군 G는 아벨군(Abelian)이며 교환법칙(Commutative)이 성립한다고 함.

갈루아 체 개론

Example

정수 집합 $\boldsymbol{Z}_m = \{0,1,\dots,m-1\}$과 모듈러 m 상의 (즉, $\bmod m$) 덧셈 연산은 항등원 0을 가지는 군을 이룸.
모든 원소 a에 대해 역원 $-a$이 존재함. 즉, $a + (-a) = 0 \bmod m$.
이 집합은 곱셈 연산에 대해서는 군을 구성하지 못함. 대부분의 원소 a가 $a \cdot a^{-1} = 1 \bmod m$인 역원을 갖지 못하기 때문임.

한 구조 내에서 기본적인 사칙연산(즉, 덧셈, 뺄셈, 곱셈, 나눗셈)이 존재하려면, 덧셈군과 곱셉군을 포함하는 집합이 필요하며, 이를 체(Field)라고 함.

Definition 4.3.2 체(Field)

- 체 F는 다음의 특성을 갖는 원소들의 집합임.
- F의 모든 원소는 군 연산 +와 항등원 0이 존재하고 교환법칙이 성립하는 덧셈아벨군(Additive Abelian Group)을 이룸.
- 0을 제외한 F의 모든 원소는 군 연산 ×와 항등원 1이 존재하고 교환법칙이 성립하는 곱셈아벨군(Multiplicative Abelian Group)을 이룸.
- 두 개의 군 연산이 결합했을 때, 분배법칙(Distributive)이 성립함. 즉, 모든 $a,b,c \in F$에 대해, $a \cdot (b+c) = a \cdot b + a \cdot c$.

갈루아 체 개론

암호기술에서는 거의 대부분 유한체 또는 갈루아 체라고 하는 유한의 원소를 가지는 체에 관심을 가지며, 체의 원소의 개수를 **Order** 또는 **Cardinality**라고 함.

Theorem 4.3.1

- 임의의 양의 정수 n과 소수(Prime Number) p에 대해 $m = p^n$일 경우에만 Order m인 체가 존재함. 이때, p를 유한체의 Characteristic이라고 함.

이 정리(Theorem)는, 예를 들어 11개의 원소 또는 $81 = 3^4$개의 원소 또는 $256 = 2^8$개의 원소를 갖는 유한체가 존재한다는 것을 의미함.
반면에 $12 = 2^2 \cdot 3$이므로 12개의 원소를 가지는 유한체는 존재하지 않는다는 것을 의미함.

소체(Prime Fields)

유한체의 가장 직관적인 예로는 소수가 Order인 체임. 예를 들아 n = 1인 체.

체 GF(p)의 원소는 정수 0,1, … , p-1로 표현되며, 체의 두 개의 연산은 모듈러 p 상에서 정수 덧셈과 정수 곱셈임.

갈루아 체 개론

Theorem 4.3.2

- 소수인 p에 대해 정수환 $\boldsymbol{Z}_p$를 $GF(p)$로 표기 가능하며 원소의 개수가 소수인 소체(Prime Field) 또는 갈루아 체(Galois Field)라고 함.
- $GF(p)$의 0이 아닌 모든 원소는 역원을 가지며, $GF(p)$의 연산은 모듈러 p 상에서 수행됨.

정수환 z_m의 경우에 대해, m이 소수이고 모듈러 덧셈 및 곱셈이 가능한 정수를 원소로 가지므로, z_m은 환일 뿐만 아니라 **유한체**이다.

소체 $GF(p)$에서 산술 연산을 수행하기 위해서는 정수환에 대한 다음의 규칙을 따라야 함.

- 덧셈과 곱셈은 모듈러 p 상에서 수행됨.
- 임의의 원소 a의 덧셈에 대한 역원은 $a + (-a) = 0 \bmod p$, 곱셈에 대한 역원은 $a \cdot a^{-1} = 1$.

Example

- $GF(5) = \{0,1,2,3,4\}$에 대한 덧셈과 곱셈의 수행 및 체에 속한 원소들 각각의 역원은 다음과 같음.

addition

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

additive inverse

$-0 = 0$
$-1 = 4$
$-2 = 3$
$-3 = 2$
$-4 = 1$

multiplication

×	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

multiplicative inverse

0^{-1} does not exist
$1^{-1} = 1$
$2^{-1} = 3$
$3^{-1} = 2$
$4^{-1} = 4$

갈루아 체 개론

AES에서 중요한 $GF(2) = \{0,1\}$에 대한 산술 연산은 다음과 같음.

addition

+	0	1
0	0	1
1	1	0

multiplication

×	0	1
0	0	0
1	0	1

위의 예제로부터 GF(2)의 (모듈러 2 상의) 덧셈은 XOR 연산과 동일하며, 곱셈은 AND 연산과 동일함을 알 수 있음.

확장체(Extension Fields) $GF(2^m)$

AES에서 유한체는 256개의 원소를 포함하며 $GF(2^8)$로 표현되며, 이는 체의 원소 각각이 하나의 바이트(Byte)로 표현 가능하기 때문임.
S-Box와 MixColumn 등에 대해 AES는 내부 데이터 경로의 각 바이트를 체 $GF(2^8)$의 원소로 다루어 유한체 내에서의 산술 연산을 수행하여 데이터를 변경함.
하지만, 유한체의 Order가 소수가 아니고, 2^8이 소수가 아니라면, 덧셈 및 곱셈 연산이 모듈러 2^8 상에서 정수의 덧셈 및 곱셈으로 표현될 수 없음.

갈루아 체 개론

$m > 1$인 그러한 체를 **확장체(Extension Field)**라고 함.
확장체를 효과적으로 다루기 위해서는 체의 원소에 대한 다른 표기법과 원소들의 산술연산에 대한 이전과는 다른 규칙이 필요함.
이를 위해, 확장체의 원소들이 **다항식(Polynomials)**으로 표현 가능하고 확장체 내에서의 계산이 특정 형태의 **다항식 산술연산(Polynomial Arithmetic)으로 계산**된다는 것을 살펴보고자 함.
확장체 $GF(2^m)$에서 원소는 정수가 아닌 $GF(2)$의 계수를 갖는 다항식으로 표현됨.
다항식의 최대 차수(Maximum Degree)는 $m-1$이므로, 각 원소에 대해 총 m개의 계수가 있음.
예를 들어, AES에서 사용되는 체 $GF(2^8)$에서 각 원소 $A \in GF(2^8)$은 다음과 같이 표현됨.

$$A(x) = a_7x^7 + a_6x^6 + \cdots + a_1x + a_0,\ a_i \in GF(2) = \{0,1\}$$

이러한 다항식은 정확하게 $256 = 2^8$개가 존재하며, 이러한 256개의 다항식의 집합이 유한체 $GF(2^8)$임.
또한, 각 다항식은 단순하게 8 bit 벡터 형태인 $A = (a_7, \ldots, a_0)$와 같이 저장될 수도 있음.

$GF(2^m)$에서의 덧셈과 뺄셈

AES의 키 덧셈 계층(Key Addition Layer)에서는 확장체에서의 덧셈을 이용함. 이를 위해 표준 다항식 덧셈 및 뺄셈을 수행함.

계수의 덧셈과 뺄셈은 GF(2)에서 수행됨.

갈루아 체 개론

Definition 4.3.3 확장체 덧셈과 뺄셈

- $A(x), B(x) \in GF(2^m)$
- Sum: $C(x) = A(x) + B(x) = \sum_{i=0}^{m-1} c_i x^i, c_i \equiv a_i + b_i \bmod 2$
- Difference: $C(x) = A(x) - B(x) = \sum_{i=0}^{m-1} c_i x^i, c_i \equiv a_i - b_i \equiv a_i + b_i \bmod 2$

계수의 덧셈과 뺄셈이 모듈러 2 상에서 수행되며, 비트 단위의 XOR 연산과 동일함을 알 수 있음.

Example

다음은 $GF(2^8)$의 두 원소의 합 $A(x) + B(x)$ 이 어떻게 계산되는지 보여줌.

$$\begin{array}{l} A(x) = x^7 + x^6 + x^4 \quad\quad + 1 \\ \underline{B(x) = \quad\quad\quad\quad x^4 + x^2 + 1} \\ C(x) = x^7 + x^6 + \quad\quad x^2 \end{array}$$

두 다항식의 차 $A(x) - B(x)$를 계산하면 두 다항식의 합 $A(x) + B(x)$과 같은 결과를 얻음을 확인할 수 있음.

갈루아 체 개론

$GF(2^m)$에서의 곱셈

$GF(2^8)$에서의 곱셈은 AES의 MixColumn 변형(Transformation)의 핵심 연산임.

곱셈 연산의 첫 단계로 유한체 $GF(2^m)$의 (다항식으로 표현되는) 두 원소는 표준 다항식 곱셈 규칙에 따라 곱해짐.

$$A(x) \cdot B(x) = (a_{m-1}x^{m-1} + \cdots + a_0) \cdot (b_{m-1}x^{m-1} + \cdots + b_0)$$

$$C'(x) = c'_{2m-2}x^{2m-2} + \cdots + c'_0$$

$$c'_0 = a_0 b_0 \bmod 2$$

$$c'_1 = a_0 b_1 + a_1 b_0 \bmod 2$$

$$\vdots$$

$$c'_{2m-2} = a_{m-1} b_{m-1} \bmod 2$$

여기에 모든 계수 a_i, b_i 및 c'_i는 $GF(2)$의 원소이고 계수의 산술 연산은 $GF(2)$에서 이루어짐에 주의해야 함.

일반적으로 곱 다항식(Product Polynomial) $C(x)$는 $m-1$보다 큰 차수를 가지게 되므로 축소되어야 함.

소체에서의 곱셈처럼 $GF(p)$에서 두 정수를 곱한 후에 그 결과를 소수로 나누고 그 나머지만을 고려함.

다항식으로 표현되는 확장체에서도 곱셈의 결과를 특정 다항식으로 나누고 그 나머지만을 고려함.

따라서 모듈러 축소를 위한 기약 다항식(Irreducible Polynomial)이 필요함.

즉, 기약 다항식은 소수에 대응되며, 기약 다항식의 인수는 1과 자기 자신의 다항식 뿐임.

갈루아 체 개론

Definition 4.3.4 확장체 곱셈

- $A(x), B(x) \in GF(2^m)$
- 기약 다항식(Irreducible Polynomial): $P(x) \equiv \sum_{i=0}^{m} p_i x^i, p_i \in GF(2)$
- Multiplication: $C(x) \equiv A(x) \cdot B(x) \bmod P(x)$

따라서, 각각의 체 $GF(2^m)$는 GF(2)의 계수와 차수가 m인 기약 다항식 P(x)가 필요함.

모든 다항식이 Irreducible한 건 아님. Ex) $x^4 + x^3 + x + 1 = (x^2 + x + 1)(x^2 + 1)$

AES에서는 기약 다항식으로 P(x) = $x^8 + x^4 + x^3 + x + 1$ 이 사용됨.

Example

체 $GF(2^4)$에서 두 개의 다항식 $A(x) = x^3 + x^2 + 1$과 $B(x) = x^2 + x$을 곱해보고자 함.
이 갈루아 체의 주어진 기약 다항식은 $P(x) = x^4 + x + 1$임.
다항식의 곱은 다음과 같이 계산됨.

$$C'(x) = A(x) \cdot B(x) = x^5 + x^3 + x^2 + x$$

갈루아 체 개론

이제 $C'(x)$를 다항식의 나눗셈 방법을 이용하여 축소함. 때로는 최고차 항 x^4와 x^5를 개별적으로 축소하는게 손쉬움.

즉, $x^4 = 1 \cdot P(x) + (x+1)$이므로 $x^4 \equiv x + 1 \bmod P(x)$이고, 따라서 $x^5 \equiv x^2 + x \bmod P(x)$.

이제 중간 결과 $C'(x)$에 x^5에 대한 축소된 표현을 넣으면 됨.

$$C(x) \equiv x^5 + x^3 + x^2 + x \bmod P(x)$$

$$C(x) \equiv (x^2+x) + (x^3 + x^2 + x) \equiv x^3$$

$$A(x) \cdot B(x) \equiv x^3$$

갈루아 체를 소프트웨어에서 구현할 때, $GF(2^m)$에서의 곱셈과 일반적인 정수 곱셈을 혼동하지 말아야 함.

다항식 즉 체의 원소는 컴퓨터의 비트 벡터로 저장되므로, 예를 들어, 다음과 같이 비트 레벨에서 계산됨에 주의해야 함.

$$\begin{aligned} A(x) \quad &\cdot \quad B(x) \quad = C(x) \\ (x^3 + x^2 + 1) &\cdot (x^2 + x) \;= x^3 \\ (1101) \quad &\cdot \quad (0110) \quad = (1000) \end{aligned}$$

갈루아 체 개론

GF(2^m)에서의 역원(Inversion)

$GF(2^8)$에서의 역원은 AES의 S-Box를 포함하는 바이트 대치 변형의 핵심 연산임.
유한체 $GF(2^m)$과 이에 해당하는 기약 다항식 $P(x)$가 주어졌을 때, 0이 아닌 $A \in GF(2^m)$의 역원 A^{-1}은 $A^{-1}(x) \cdot A(x) = 1 \bmod P(x)$와 같이 정의됨.
작은 사이즈의 체(보통 2^{16} 보다 작은 경우)에 대해서는 미리 계산된 역원을 포함하는 Lookup Table을 이용하여 구함(Table 4.2 참조).
Table 4.2는 AES의 S-Box에서 사용되는 값을 보여주며, $GF(2^8) \bmod P(x)$에 대한 모든 역원을 16진법으로 나타내고 있음.
단, 체 원소 0은 역원이 존재하지 않기에, 특별한 경우로 분류하여 입력값 0에 대한 결과값을 0으로 대응함.

Table 4.2 Multiplicative inverse table in $GF(2^8)$ for bytes xy used within the AES S-Box

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

갈루아 체 개론

Example

주어진 값이 16진수로 C2라면 $x^7 + x^6 + x = (11000010)_2 = (C2)_{16} = (XY)$라고 표현 가능함.
이 수의 역원을 Table 4.2에서 찾아보면 $(2F)_{16} = (00101111)_2 = x^5 + x^3 + x^2 + x + 1$임.
이는 다음의 곱셈을 통해 검증 가능함.

$$(x^7 + x^6 + x) \cdot (x^5 + x^3 + x^2 + x + 1) \equiv 1 \bmod P(x)$$

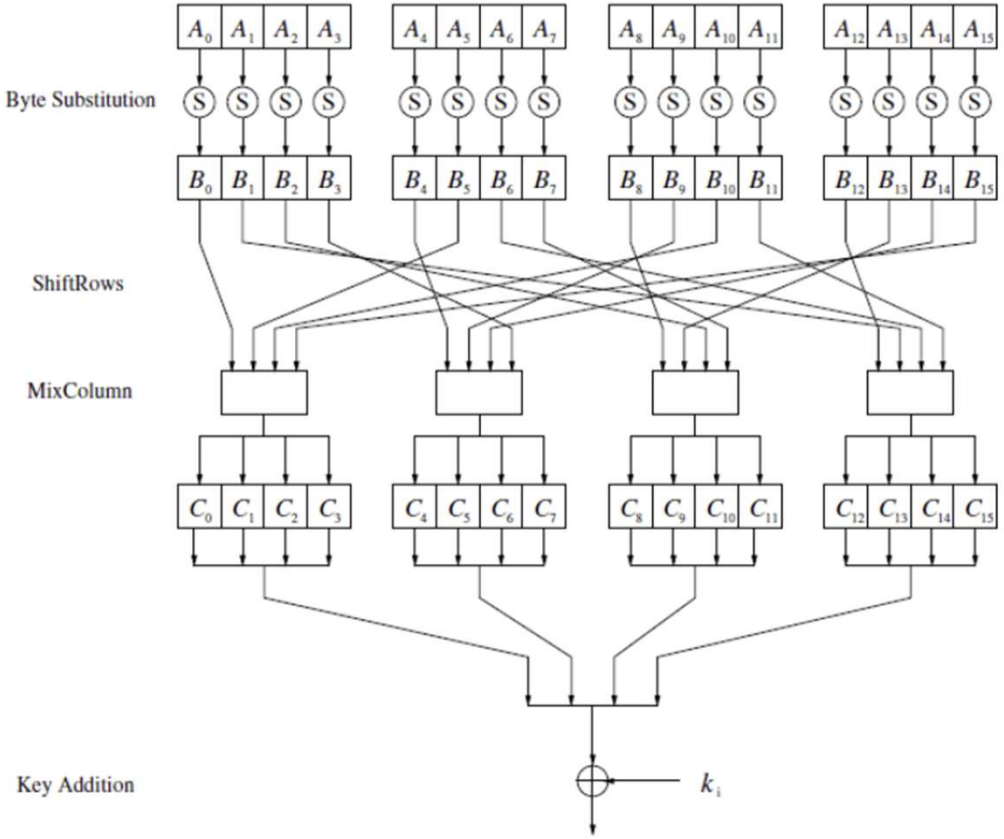
Table 4.2 Multiplicative inverse table in $GF(2^8)$ for bytes xy used within the AES S-Box

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

AES의 내부 구조

AES의 내부 구조

$A_0, \ldots, A_{15}$: 16-byte Input (즉, A_i: 1 byte)
$B_0, \ldots, B_{15}$: 16-byte Output of S-Box
$C_0, \ldots, C_{15}$: 16-byte Output of ShiftRows and MixColumn Transformations
k_i: 128-bit Subkey.
$C_0, \ldots, C_{15}$와 k_i가 XOR 연산됨.



AES에서 데이터 상태(State)

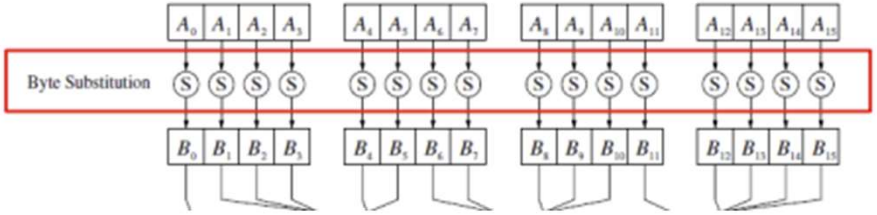
16-byte(128-bit) Data Path: 4 × 4 행렬

A_0	A_4	A_8	A_{12}
A_1	A_5	A_9	A_{13}
A_2	A_6	A_{10}	A_{14}
A_3	A_7	A_{11}	A_{15}

AES의 내부 구조

바이트 대치 계층(Byte Substitution Layer)

바이트 대치 계층은 다음의 특성을 갖는 16개의 **S-Box**로 구성됨.



모든 16개의 S-Box는 동일함(Identical)

AES의 유일한 비선형(Nonlinear) 요소임.

le 4.3 AES S-Box: Substitution values in hexadecimal notation for input byte (xy)

	y															
	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
x 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

AES의 내부 구조

Table 4.3 AES S-Box: Substitution values in hexadecimal notation for input byte (xy)

	y															
	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
x 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

Example

S-Box의 입력 $A_i = (C2)_{16}$에 대해 S-Box 출력 결과는 Table의 결과에 따라 $S((C2)_{16}) = (25)_{16}$임.

비트 단위에서는 다음과 같이 표현 → S(11000010) = (00100101)

$S(A_i) = A_i$인 A_i가 존재하지는 않음. ex) S(00000000) = (01100011)

AES의 내부 구조

확산 계층(Diffusion Layer)

모든 입력 상태 비트에 확산(Diffusion)을 제거하기 위한 계층.

두 개의 하위 계층(Sublayer)로 구성.

ShiftRows Sublayer : 바이트 단위의 데이터 전치(Permutation).

MixColumn Sublayer : 4 바이트 블록을 결합. → 행렬 연산

확산 계층에서는 비선형 연산인 S-Box와는 다르게 상태(State) 행렬 A,B에 대해 선형 연산 진행.

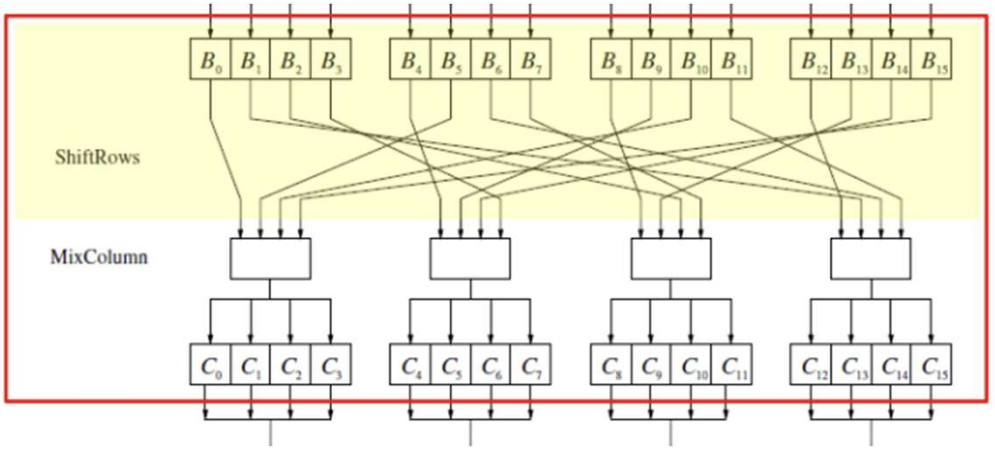
즉, $\mathrm{DIFF}(\boldsymbol{A}) + \mathrm{DIFF}(\boldsymbol{B}) = \mathrm{DIFF}(\boldsymbol{A} + \boldsymbol{B})$

AES의 내부 구조

확산 계층(Diffusion Layer) – ShiftRows 하위 계층

Input Matrix $\boldsymbol{B} = (B_0, \ldots, B_{15})$

B_0	B_4	B_8	B_{12}
B_1	B_5	B_9	B_{13}
B_2	B_6	B_{10}	B_{14}
B_3	B_7	B_{11}	B_{15}



Output Matrix

B_0	B_4	B_8	B_{12}	no shift
B_5	B_9	B_{13}	B_1	← one position left shift
B_{10}	B_{14}	B_2	B_6	← two positions left shift
B_{15}	B_3	B_7	B_{11}	← three positions left shift

AES의 내부 구조

확산 계층(Diffusion Layer) – MixColumn 하위 계층

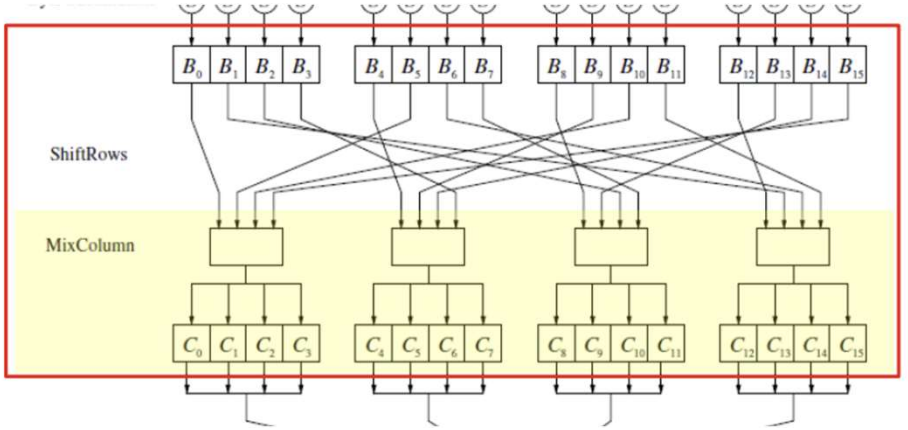
$$\text{MixColumn}(\boldsymbol{B}) = \boldsymbol{C}$$

4 byte의 각 열을 벡터로 간주하고 고정된 4×4 행렬과 곱함 (Matrix Multiplication).

- 예를 들어, 첫 4개의 출력 바이트는 다음과 같이 계산됨

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{pmatrix} \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix}$$

모든 산술 연산은 **갈루아 체 $GF(2^8)$에서 수행됨**.



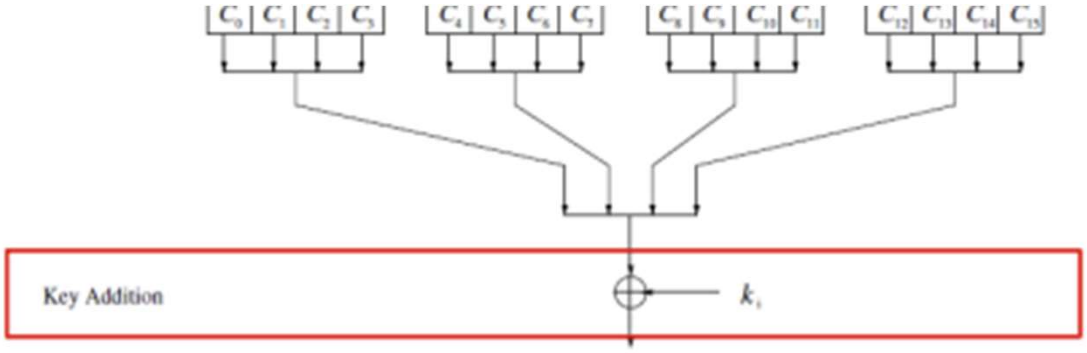
AES의 내부 구조

키 덧셈 계층(Key Addition Layer)

입력(Input)
- 16-byte 상태 행렬 C
- 16-byte 서브키 k_i

출력(Output)
- $C \oplus k_i$ (i.e., XOR 연산)

서브키들은 키 스케줄(Key Schedule)에 의해 생성됨.



키 스케줄(Key Schedule)

서브키들은 128, 192, 256 bit의 오리지널 입력 키(Original Input Key)로 부터 재귀적으로 유도됨.
AES 수행을 시작할 때 1개의 서브키가 필요하며, 각 라운드 별로 1개의 서브키가 필요함.
- 따라서, 총 서브키의 개수는 $N_r + 1$임.
- 예를 들어, 192 bit AES의 경우 12 라운드로 구성되어 있으므로 13개의 서브키가 필요함.

AES의 내부 구조

128-bit 키를 갖는 AES의 키 스케줄

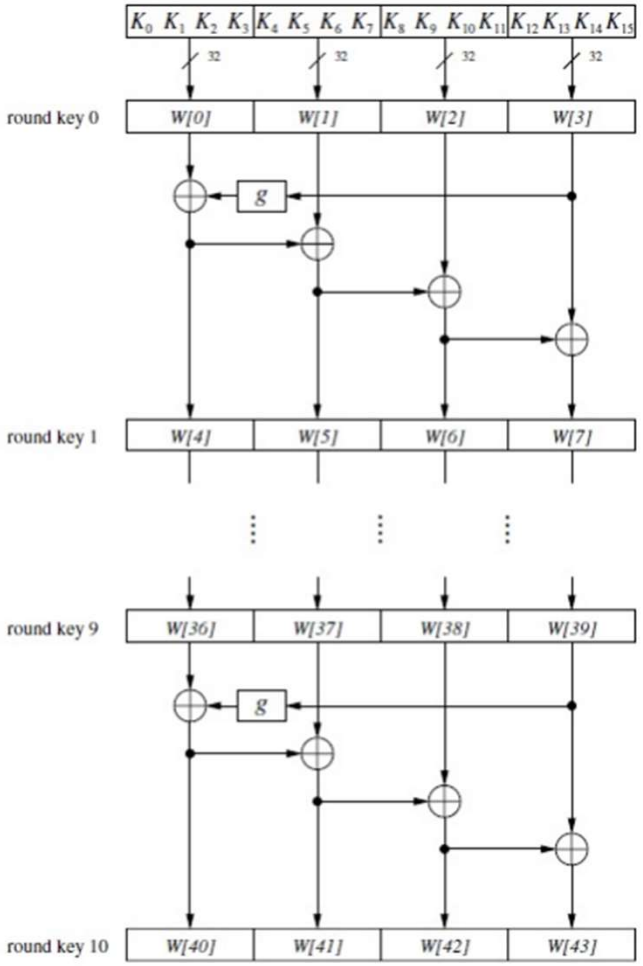
Word 단위의 연산을 수행 (1 word = 32 bits)
11개의 서브키가 $W[0], \dots, W[3], \dots, W[40], \dots W[43]$에 저장됨.
$K_0, \dots, K_{15}$ 는 오리지널 AES 키의 각 바이트를 나타냄.
첫 번째 서브키 k_0는 오리지널 AES 키임.
일반적인 서브키$(k_1, \dots k_{10})$에서 가장 왼쪽 (Leftmost)의 Word$(W[4], W[8], \dots, W[40])$는 다음과 같이 계산되며,

$$W[4i] = W[4(i-1)] + g(W[4i-1])$$

이어진 나머지 세 Words는 다음과 같이 계산됨.

$$W[4i+j] = W[4i+j-1] + W[4(i-1)+j]$$

for $i = 1, \dots, 10,\ j = 1,2,3.$



AES의 내부 구조

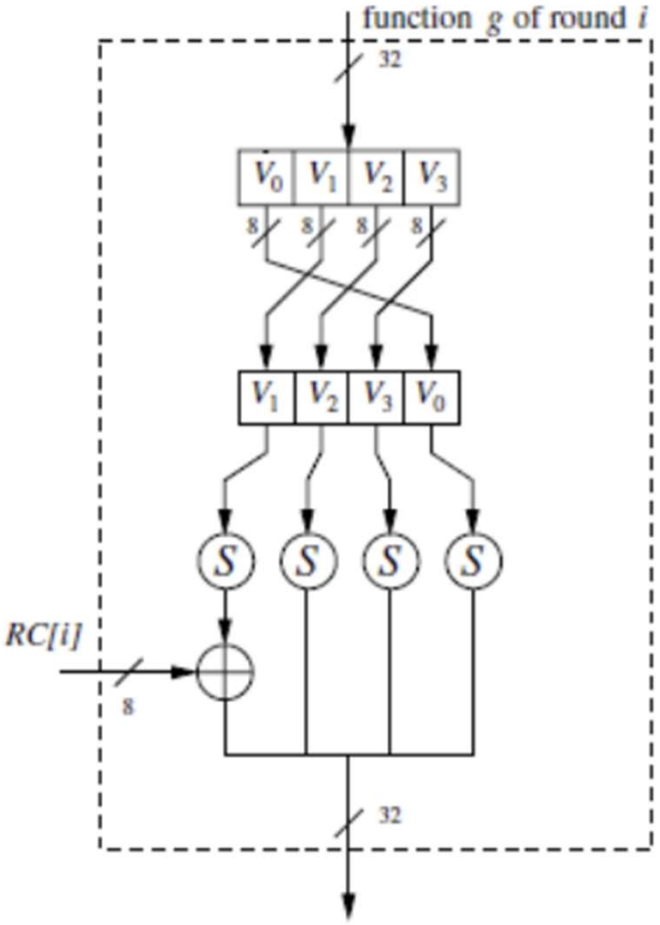
128-bit 키를 갖는 AES의 키 스케줄

함수 $g(\cdot)$는 4개의 입력 바이트를 순환 이동하고 바이트 단위의 S-Box 대치를 수행함.
또한, 라운드 계수(RC, Round Coefficient)를 그 결과 일부에 더함.
8 bit 값의 RC는 갈루아 체 $GF(2^8)$내의 원소를 나타내며, 각 라운드 마다 다양한 값을 가짐.

$$
\begin{aligned}
RC[1] &= x^0 = (00000001)_2, \\
RC[2] &= x^1 = (00000010)_2, \\
RC[3] &= x^2 = (00000100)_2, \\
&\vdots \\
RC[10] &= x^9 = (00110110)_2.
\end{aligned}
$$

$g(\cdot)$는 다음의 두 가지 목적을 가짐.

- 키 스케줄에 비선형성(Nonlinearity)을 더함.
- AES에서의 대칭성(Symmetry)를 제거함.



Q & A