제 6 장 대칭 암호알고리즘: AES

6.1 AES

6.1.1 AES 개요

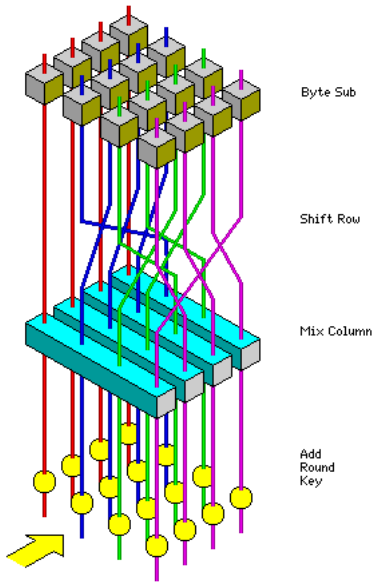
1977년도에 미국 표준으로 제정된 DES는 지금까지 큰 허점이 발견되지 않았지만 키 길이가 56비트 밖에 되지 않아 현재의 컴퓨팅 기술로는 쉽게 전사공격을 하여 암호해독을 할 수 있다. 따라서 1997년에 새 표준에 대한 작업을 시작하여 2000년 10월에 AES(Advanced Encryption Standard)라는 새 표준을 채택하였다. 1997년 새 표준에 대한 제안에 의하면 새 암호알고리즘의 블록 크기는 128비트이어야 하며, 알고리즘에 대한 변경 없이 128비트, 196비트, 256비트 길이의 키를 지원해야 한다. 1998년도에 제출된 여러 제안 중에 15개를 일차적으로 선정하였고, 1999년에 이 중에 다섯 개를 최종 후보로 선정하였다. 이 중에 벨기에 암호학자인 Daemen과 Rijmen이 제안한 Rijndael 암호알고리즘이 AES로 채택되었다.

채택된 Rijndael 알고리즘은 기존 표준인 DES와 달리 Feistel 구조가 아니었으며, 제안에서 요구한 사항뿐만 아니라 블록 크기를 192비트, 256비트로 확장할 수 있도록 되어 있었다. 하지만 제안에서 요구한 128비트 블록 크기 버전만 표준으로 채택되었다. Rijndael 암호알고리즘의 라운드 수는 블록 크기와 키 길이에 의해 다음과 같이 결정된다. N_B가 암호블록의 크기에 대한 32비트 워드의 수이고, N_K가 암호키 길이에 대한 32비트 워드의 수이면 라운드의 수 $N_R = 6 + \max(N_B, N_K)$이다. 표준으로 채택된 블록의 길이는 128비트이므로 $N_B = 4$이며, 128비트, 192비트, 256비트 세 가지 종류의 키 길이를 지원하므로 $N_K = 4, 6, 8$이다. 따라서 블록 크기가 128비트이고, 키 길이가 128비트이면 $N_R = 10$이 된다. 이 때 암호키는 4행×N_K 2차원 바이트 배열로 구성된 것으로 간주한다. 이 암호키는 $4N_R + 4$개의 32비트 워드로 확장되어 사용된다. 즉, 블록 크기와 키 길이가 모두 128비트이면 128비트 암호키는 총 44개의 32비트 워드로 확장되며, 각 라운드마다 이 중 4개의 워드가 사용된다.

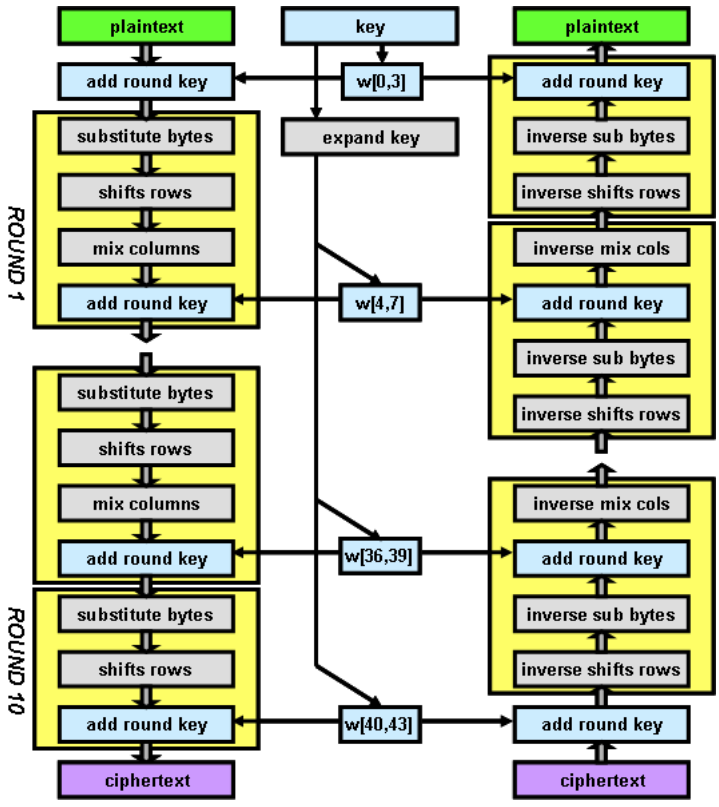
각 라운드는 그림 6.1처럼 하나의 자리바꿈 연산과 세 개의 치환 연산으로 구성되어 있다. 그림에서 알 수 있듯이 AES의 라운드는 매우 단순하다.

- S-박스: $GF(2^8)$을 이용한 치환연산
- 행이동(shift row): 단순 자리바꿈
- 열섞음(mix column): $GF(2^8)$을 이용한 치환연산
- 라운드키 적용(add roundkey): XOR 연산을 이용

여기서 $GF(2^8)$을 이용한 계산이란 기약다항식 $m(x) = x^8 + x^4 + x^3 + x + 1$을 사용하는 다항식 체를 말한다. 위 네 가지 세부 연산은 모두 역이 가능하다. 따라서 라운드 키를 적용하는 부분을 제외하고는 그 자체만으로는 어떤 안전성도 제공하지 못한다. 다른 대부분의 알고리즘과 마찬가지로 복호화는 키를 역순으로 사용하며, 전체 알고리즘은 그림 6.2와 같다.



<그림 6.1> AES의 한 라운드 구성도

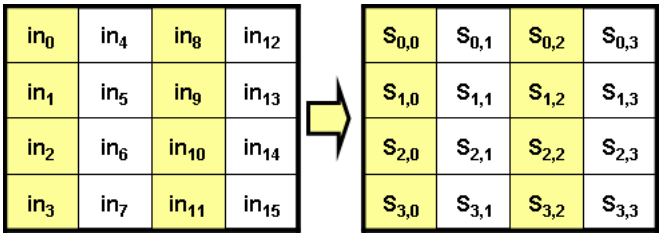


<그림 6.2> AES 알고리즘

그림 6.2에서 알 수 있듯이 복호화에서 사용되는 연산은 암호화에서 사용된 연산의 역 연산이다. 앞서 언급한 바와 같이 각 라운드에서 사용되는 모든 연산은 역이 가능하다. 따라서 그 역을 이용하여 복호화가 이루어진다.

6.1.2 상태

AES의 모든 연산들은 상태(state)라고 하는 2차원 바이트 배열에 수행된다. 이 상태는 항상 4행으로 구성되며, 각 행은 N_B 바이트로 구성된다. 하지만 표준에서 $N_B = 4$이므로 표준에서 사용하는 상태는 항상 4×4 바이트 행이 된다. 128비트 입력을 상태로 전환하는 방법은 그림 6.3과 같다.



<그림 6.3> AES에서 입력을 상태로 변환하는 방법

예를 들어 128 비트 입력이 EA835CF00445332D655D98AD8596B0C5와 같으면 이것의 상태는 그림 6.4와 같다.

EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

<그림 6.4> AES의 상태의 예

6.1.3 S-Box 치환

암호화 과정의 각 라운드에서 가장 먼저 수행되는 연산은 s-box 치환 연산이다. 이 연산은 상태를 구성하는 각 바이트를 s-box을 이용하여 치환한다. 두 개의 s-box가 있으며, 하나를 전방향 s-box라 하고, 다른 하나를 역방향 s-box라 한다. 두 개의 s-box은 각각 그림 6.5와 6.6에 기술되어 있다. 두 s-box는 서로 역 관계에 있다. 즉, 특정 바이트 값을 전방향 s-box으로 치환한 후에 그 결과를 다시 역방향 s-box로 치환하면 원래 값을 얻게 된다. 따라서 전방향 s-box는 암호화할 때 사용되고, 역방향 s-box는 복호화할 때 사용된다.

AES에 사용되는 전방향 s-box는 다음과 같은 원리에 의해 구성되었다.

- **단계 1.** s-box를 00, 01, ..., FF 순으로 초기화한다.
- **단계 2.** 이 값들을 $GF(2^8)$에서 곱셈에 대한 역원으로 매핑한다. 단, 곱셈에 대한 역원이 없는 00은 00으로 매핑한다.
- **단계 3.** 한 항의 값이 $b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$이면 다음과 같이 변형한다. 여기서 주의해야 할 것은 행렬 곱셈이나 덧셈 모두 XOR 연산이 수행된다.

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \\ 11111000 \\ 01111100 \\ 00111110 \\ 00011111 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

예6.1) 95에 해당되는 s-box의 값은?

- 95의 곱셈에 대한 역원은 8A이다. 그러면 b'_0은 다음과 같이 계산된다.

  (10001111 × 01010001) $\oplus$ 1 = (1$\oplus$1$\oplus$0$\oplus$1$\oplus$1$\oplus$1$\oplus$1$\oplus$1$\oplus$0)$\oplus$1 = 1$\oplus$1 = 0
- 이와 같이 계산하면 결과는 01010100 = 2A가 된다.

S-box는 알려진 공격에 안전하도록 설계되었으며, $s-box(a)=a$가 되는 경우 또는 $s-box(a)=\overline{a}$가 없도록 설계되었다. 또한 그 역을 취할 수 있도록 만들었다,

	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

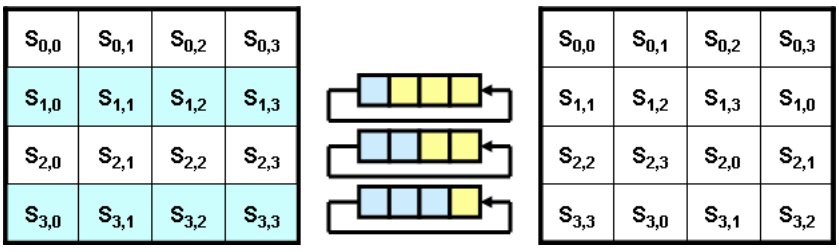
<그림 6.6> AES의 전방향 s-box

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

<그림 6.7> AES의 역방향 s-box

6.1.4 행이동 자리바꿈

AES의 각 라운드에 사용되는 4개의 기본 연산 중 나머지 세 연산은 치환이고, 이 연산만 자리바꿈이다. 이 연산은 그림 6.8과 같이 이루어지며, 복호화할 때에는 그림 6.8과 정반대로 이루어진다. 이 연산은 자리바꿈을 통해 암호화 과정이 평문에 모든 비트에 고루 영향을 주도록 하기 위함이다. 더욱이 평문 비트들이 상태로 전환되어 열 단위로 연산을 적용받기 때문에 이와 같은 이동이 영향을 분산시키는 효과는 매우 크다.



<그림 6.8> AES의 행이동 자리바꿈 연산

6.1.5 열섞음 치환

이 연산은 상태 행렬을 다음 행렬에 곱하여 값을 치환하게 되며, 이 때 곱셈 연산은 $GF(2^8)$에서 계산된다.

$$\begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} [S] = [S']$$

이 계산은 두 가지 형태로 계산될 수 있다. 첫째는 실제 행렬 곱셈을 다음과 같이 하는 것

이다.

$$s'_{0,0} = (02 \bullet s_{0,0}) \oplus (03 \bullet s_{1,0}) \oplus (03 \bullet s_{2,0}) \oplus (03 \bullet s_{3,0})$$

이 경우 이와 같은 계산을 총 16번 해야 한다. 둘째는 입력 상태의 열을 다음과 같은 3차 다항식으로 생각하고

$$s_0(x) = s_{3,0}x^3 + s_{2,0}x^2 + s_{1,0}x + s_{0,0}$$

법 x^4+1에서 $a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$에 곱하는 것이다. 이렇게 하여 얻어진 3차 다항식의 각 계수는 출력 상태의 열이 된다.

예6.2) 다음은 주어진 입력 상태를 열섞음 연산에 적용하였을 때 그 결과를 보여주고 있다.

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

⇨

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

이 예에서 $s'_{0,0}$이 47이 되는 과정을 살펴보면 다음과 같다.

$$(\{02\} \times \{87\}) \oplus (\{03\} \times \{6E\}) \oplus \{46\} \oplus \{A6\} = \{47\}$$

- 02와 87를 다항식으로 표현하면 각각 x와 x^7+x^2+x+1와 같다. 두 값을 법 $x^8+x^4+x^3+x+1$에서 곱하면 그 결과는 x^4+x^2+1이 된다. 따라서 이것을 다시 비트로 표현하면 0001 0101이 된다.

그러면 왜 두 가지 방식의 계산이 동일한 결과를 얻는지 살펴보면 다음과 같다. 계수가 $GF(2^8)$인 두 개의 4차 다항식을 법 x^4+1에서 곱한 결과는 다음과 같다.

- $a(x) = a_3x^3 + a_2x^2 + a_1x + a_0,\ b(x) = b_3x^3 + b_2x^2 + b_1x + b_0$
- $a(x) \times b(x) = c(x) = c_6x^6 + c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1 + c_0$
  - $c_0 = a_0 \times b_0$
  - $c_1 = (a_1 \times b_0) \oplus (a_0 \times b_1)$
  - $c_2 = (a_2 \times b_0) \oplus (a_1 \times b_1) \oplus (a_0 \times b_2)$
  - $c_3 = (a_3 \times b_0) \oplus (a_2 \times b_1) \oplus (a_1 \times b_2) \oplus (a_0 \times b_3)$
  - $c_4 = (a_3 \times b_1) \oplus (a_2 \times b_2) \oplus (a_1 \times b_3)$
  - $c_5 = (a_3 \times b_2) \oplus (a_2 \times b_3)$
  - $c_6 = a_3 \times b_3$

  이 다항식을 x^4+1로 나누었을 때 나머지를 구하면 다음과 같다.
- $d(x) = c(x) \bmod x^4+1 = d_3x^3 + d_2x^2 + d_1x + d_0$
  - $d_0 = (a_0 \times b_0) \oplus (a_3 \times b_1) \oplus (a_2 \times b_2) \oplus (a_1 \times b_3)$
  - $d_1 = (a_1 \times b_0) \oplus (a_0 \times b_1) \oplus (a_3 \times b_2) \oplus (a_2 \times b_3)$

- $d_2 = (a_2 \times b_0) \oplus (a_1 \times b_1) \oplus (a_0 \times b_2) \oplus (a_3 \times b_3)$
- $d_3 = (a_3 \times b_0) \oplus (a_2 \times b_1) \oplus (a_1 \times b_2) \oplus (a_0 \times b_3)$

따라서 계수가 $GF(2^8)$인 두 개의 4차 다항식을 법 x^4+1에서 곱한 결과는 다음과 같은 행렬식으로 표현이 가능하다.

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} a_0\ a_3\ a_2\ a_1 \\ a_1\ a_0\ a_3\ a_2 \\ a_2\ a_1\ a_0\ a_3 \\ a_3\ a_2\ a_1\ a_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

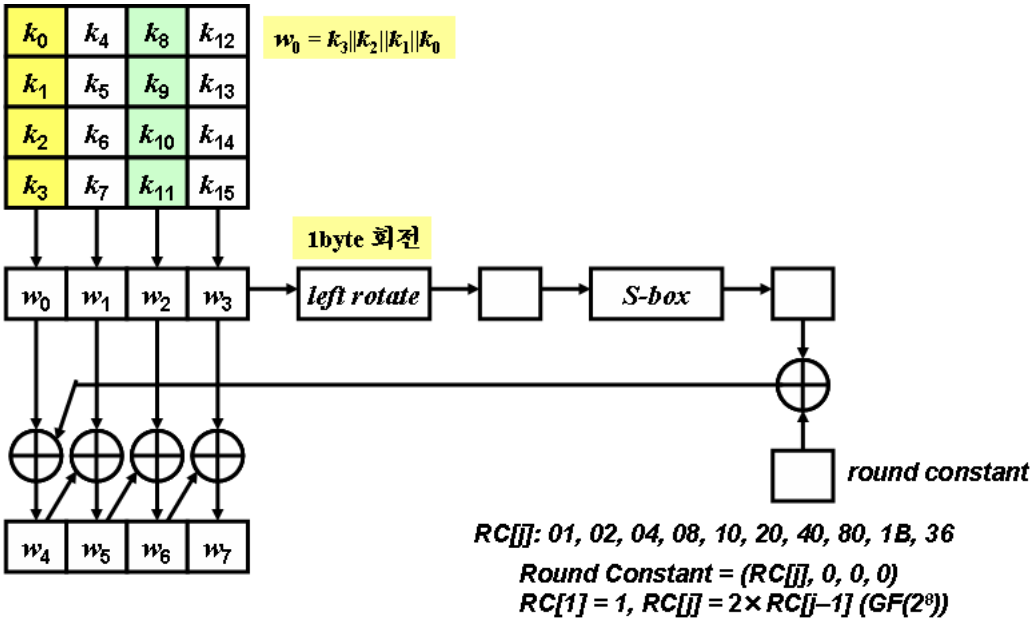
따라서 $a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$이면 행렬식은 다음과 같이 된다.

$$\begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} [S] = [S']$$

이 과정을 역하기 위해서는 사용된 곱셈 행렬의 역행렬만 있으면 된다. 따라서 복호화할 때에는 다음과 같은 행렬을 사용한다.

$$\begin{bmatrix} 0E\ 0B\ 0D\ 09 \\ 09\ 0E\ 0B\ 0D \\ 0D\ 09\ 0E\ 0B \\ 0B\ 0D\ 09\ 0E \end{bmatrix} [S] = [S']$$

6.1.6 AES 키 스케줄링



<그림 6.10> AED 키 스케줄링 알고리즘

AES는 사용되는 암호키의 길이가 128비트이면 총 44개의 32비트 워드로 확장되어 각 라운드마다 4개의 32비트 워드를 라운드 키로 사용한다. AES의 키 스케줄링 알고리즘은 그림

6.10과 같으며 다음과 같이 진행된다.

- **단계 1.** 128비트 키를 그림 4.7에 기술된 것처럼 4개의 32비트 워드로 바꾼다. 이 3개의 값이 첫 4개의 32비트 워드가 된다.
- **단계 2.** 첫 4개의 워드 중 마지막 워드는 1 바이트 왼쪽 순환 이동된 뒤에 전방향 s-box를 이용하여 치환된다. 그 다음에 라운드 상수와 XOR된다. 결과 값은 첫 워드와 XOR되어 다음 4개 워드의 첫 워드가 만들어진다.
- **단계 3.** 이렇게 만들어진 첫 워드와 기존 4개의 워드 중 두 번째 워드가 XOR되어 두 번째 워드가 만들어지고, 이 결과와 세 번째 워드가 XOR되어 세 번째 워드가 만들어지며, 마지막으로 이 결과와 네 번째 워드가 XOR되어 마지막 워드가 만들어진다.
- **단계 4.** 단계 2부터 3을 9번 수행하여 각 라운드 키를 생성한다.

참고문헌

[1] Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, Nov. 2001.