

민관겸용 블록 암호 알고리즘

ARIA

알고리즘 명세서

목 차

1	서론	4
2	정의	4
	2.1 용어 와 약어	4
	2.2 알고리즘 파라미터, 기호, 함수	5
3	표기와 규약	5
	3.1 입력과 출력	5
	3.2 바이트	6
	3.3 바이트 배열	7
	3.4 상태(State)	7
4	수학적 배경 지식	7
	4.1 덧셈 연산	8
	4.2 곱셈 연산	8
5	알고리즘 구조	9
	5.1 치환 계층	13
	5.2 확산 계층	14
	5.3 키 확장	17
6	암/복호화 예제	21
	6.1 키 확장 : 128 비트 암호키	21
	6.2 키 확장 : 192 비트 암호키	23
	6.3 키 확장 : 256 비트 암호키	25
	6.4 암/복호화 예제 : 128 비트 암호키	27
	6.5 암/복호화 예제 : 192 비트 암호키	31
	6.6 암/복호화 예제 : 256 비트 암호키	35
	참고 문헌	40

표 목차

1	비트 패턴의 16진법 표현	6
2	바이트와 비트의 색인	7
3	ARIA 사양	9
4	S-box S_1	15
5	S-box S_1^{-1}	15
6	S-box S_2	16
7	S-box S_2^{-1}	16
8	암호키 길이에 따른 초기화 상수	18
9	키 확장을 위한 중간값: 128 비트	21
10	키 확장 초기화: 128 비트	22
11	키 생성 — 128 비트	22
12	키 확장을 위한 중간값: 192 비트	23
13	키 확장 초기화: 192 비트	23
14	키 생성 — 192 비트	24
15	키 확장을 위한 중간값: 256 비트	25
16	키 확장 초기화: 256 비트	25
17	키 생성 — 256 비트	26
18	jARIA-128 암호화 예제 벡터	27
19	jARIA-128 복호화 예제 벡터	29
20	jARIA-192 암호화 예제 벡터	31
21	jARIA-192 복호화 예제 벡터	33
22	jARIA-256 암호화 예제 벡터	35
23	jARIA-256 복호화 예제 벡터	37

그림 목차

1	ARIA의 Pseudo Code	10
2	암호화 및 복호화 과정	11
3	키 확장 과정을 통한 암호화 과정	12
4	두 유형의 치환 계층	13
5	키 확장 초기화 과정	18
6	ARIA 키 확장의 Pseudo Code	19

1 서론

이 문서에서는 블록 암호 ARIA를 자세히 서술한다. 블록 암호 ARIA는 128-비트 데이터 블록을 처리하는 알고리즘으로 128, 192, 256 비트 암호키를 사용한다.

ARIA는 위의 세 가지 길이의 암호키로 사용될 수 있으며 요구되는 안전성 기준에 따라 용도가 구분될 수 있다. 이 문서에는 암호키의 길이에 따라 **ARIA-128**, **ARIA-192**, **ARIA-256**으로 구분하여 표기한다.

이후 절들은 다음과 같은 내용으로 구성되어 있다.

- 용어 정의, 약어, 알고리즘 파라미터, 기호, 함수
- 알고리즘 명세서에 사용된 표기, 규약
- 알고리즘 설명에 사용된 수학적 배경 지식
- 키 확장, 암호화/복호화 과정을 서술한 알고리즘 명세서
- 암호/복호화 예제

2 정의

2.1 용어 와 약어

이 문서에서는 다음의 용어와 약어를 사용한다.

ARIA	Academy, Research Institute, Agency의 약어로 학·연·관이 공동으로 개발한 정보보호의 핵심 기술임을 함축하고 있음
Involution 구조	암호화 과정과 복호화 과정이 같은 구조
S-box	비선형 치환 테이블로 바이트 치환에 사용됨
SPN 구조	Substitution-Permutation-Networks 구조로 S-box와 확산 함수가 반복적으로 사용되는 구조
Feistel 구조	데이터를 두 블록으로 나누어 좌·우부분에 교대로 비선형 변환을 적용시키는 구조
대칭키 암호	암·복호화 키가 같은 암호
라운드 키	암호키로부터 키 확장을 통하여 생성되는 값들로 암호화 및 복호화 상태에 적용됨
라운드 함수	블록 암호의 각 라운드에서 사용되는 함수
바이트	여덟 비트 군으로 단일 개체 또는 여덟 비트의 배열로 다루어진다.
배열	바이트 등의 순차적 모임

복호화	암호키를 이용하여 암호문을 평문으로 바꾸는 일련의 변환들
블록	입력, 상태, 출력, 라운드 키를 구성하는 비트 열로 열의 길이는 포함하는 비트 수를 표시, 블록은 바이트의 배열로도 해석됨
블록 암호	고정된 길이의 평문 블록을 고정된 길이의 암호문 블록으로 변환하는 함수
비트	0 또는 1의 값을 갖는 한자리 숫자.
상태	암호화, 복호과정의 중간 값
아핀 변환	행렬 곱과 벡터 합이 순차적으로 구성된 변환
암호화	암호키를 이용하여 평문을 암호문으로 바꾸는 일련의 변환들
암호문	평문을 합당하게 푸는 방법을 모르는 사람에게는 알 수 없는 형태로 변환한 데이터
암호키	평문 또는 암호문의 암호화, 복호화에 사용되는 비밀정보
키 확장	암호키로 부터 라운드 키들을 생성하는 과정
평문	암호화되지 않은 원래의 정보

2.2 알고리즘 파라미터, 기호, 함수

이 문서에서는 다음과 같은 알고리즘 파라미터, 기호, 함수들이 사용된다.

AddRoundKey()	라운드 키를 상태와 XOR하는 암호화, 복호화 과정의 변환
SubstLayer()	S-box와 이의 역변환으로 이루어진 계층
DiffLayer()	상태의 바이트들을 섞는 암호화, 복호화 과정의 변환 계층
MK	암호키
XOR	배타적 논리합 연산
\oplus	배타적 논리합 연산
$A \lll k$	A 의 각 비트를 왼쪽으로 k 비트씩 순환이동
$A \ggg k$	A 의 각 비트를 오른쪽으로 k 비트씩 순환이동

3 표기와 규약

3.1 입력과 출력

블록 암호 ARIA의 입력과 출력은 각각 128비트들의 수열로 구성된다. 이때 블록 길이는 이 나열들에서 블록에 포함된 비트 수를 의미한다. 블록 암호 ARIA의 암호키는 128, 192 또는 256 비트들의 수열이다.

수열에서의 비트들은 0에서부터 시작하여 수열 길이(블록 길이 또는 키 길이)보다

하나 작은 수에서 끝나도록 번호를 붙인다. 비트에 부여된 수 i 는 비트 색인이며 블록 길이와 키 길이에 따라 $0 \leq i < 128, 0 \leq i < 192, 0 \leq i < 256$ 중 하나의 범위에 있다.

3.2 바이트

블록 암호 ARIA의 기본 연산 단위는 바이트(여덟 비트의 수열)이다. 입력·출력과 암호 키의 비트 수열은 연속된 여덟 바이트들로 묶은 바이트의 배열로 3.3절에서와 같이 처리된다.

a 를 입력, 출력 또는 암호키라고 할 때, 해당 바이트 배열은 a_n 또는 $a[n]$ 으로 표시된다. n 은 다음과 같은 범위에 있다.

키 길이 = 128 비트, $0 \leq n < 16$; 블록 길이 = 128 비트, $0 \leq n < 16$;

키 길이 = 192 비트, $0 \leq n < 24$;

키 길이 = 256 비트, $0 \leq n < 32$;

블록 암호 ARIA의 모든 바이트 값은 괄호 $\{b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0\}$ 안의 각 비트 값 (0 또는 1)의 연결로 표현된다. 바이트들은 다음과 같은 다항식 표현을 통하여 유한체 $GF(2^8)^1$ 원소로 해석할 수도 있다.

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 = \sum_{i=0}^7 b_i x^i.$$

예들 들면, $\{01100011\}$ 은 유한체 원소로 표현하면 $x^6 + x^5 + x + 1$ 이 된다. 또한, 바이트 값은 16진법을 이용한 표기가 많이 이용되는데, 이 표기는 바이트의 여덟 비트를 네 비트 씩 두 군으로 묶고 네 비트를 다음과 같이 단일 문자에 대응시켜 표기한다.

<표 1> 비트 패턴의 16진법 표현

비트 패턴	문자	비트 패턴	문자	비트 패턴	문자	비트 패턴	문자
0000	0	0100	4	1000	8	1100	c
0001	1	0101	5	1001	9	1101	d
0010	2	0110	6	1010	a	1110	e
0011	3	0111	7	1011	b	1111	f

$\{01100011\}$ 의 경우 위 표를 이용하면 0110은 6, 0011은 3에 대응되므로 16진법으로 $\{63\}$ 으로 표현된다. $\{63\}$ 은 0x63 표기하기도 한다.

¹계수가 0 또는 1을 갖는 다항식의 집합 $\mathbb{Z}_2[x]$ 를 8차 기약다항식 $m(x)$ 로 나눈 집합

3.3 바이트 배열

비트 배열로부터 바이트 배열을 얻는 방법은 다음과 같다.

128 비트 입력 수열 $input_0 input_1 input_2 \cdots input_{127}$ 로부터 바이트 배열 $a_0 a_1 a_2 \cdots a_{15}$ 은 다음과 같이 주어진다.

$$\begin{aligned} a_0 &= \{input_0, input_1, \cdots, input_7\}; \\ a_1 &= \{input_8, input_9, \cdots, input_{15}\}; \\ &\vdots \\ a_{15} &= \{input_{120}, input_{121}, \cdots, input_{127}\} \end{aligned}$$

192 비트, 256 비트 입력 수열에 대해서도 위 정렬 방식이 적용되는데 $n + 1$ 번째 바이트는 다음과 같이 주어진다.

$$a_n = \{input_{8n}, input_{8n+1}, \cdots, input_{8n+7}\}$$

3.2절과 3.3절의 표기를 종합하면 다음과 같다.

<표 2> 바이트와 비트의 색인

입력 비트 수열	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
바이트 색인	0								1								...
바이트 내 비트 색인	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	...

3.4 상태(State)

내부적으로, 블록 암호 ARIA의 각 연산은 상태(State)로 불리는 바이트 배열에 대하여 수행된다. 상태는 16개의 바이트로 구성된다. 상태에서 바이트 및 비트 정렬은 3.3절의 블록 내에서의 정렬 방식을 따른다.

4 수학적 배경 지식

블록 암호 ARIA의 바이트들은 3.2절의 표기를 이용하여 유한체 $GF(2^8)$ 원소로 해석될 수 있다. 바이트들의 덧셈 및 곱셈은 유한체에서의 곱셈 및 덧셈으로부터 유도된다.

4.1 덧셈 연산

유한체 원소의 덧셈은 바이트 내에서 대응되는 비트들의 모듈로 2 연산으로 서술된다. 두 바이트 $\{a_7a_6a_5a_4a_3a_2a_1a_0\}, \{b_7b_6b_5b_4b_3b_2b_1b_0\}$ 에 대하여 합을 $\{c_7c_6c_5c_4c_3c_2c_1c_0\}$ 라 하면 $c_i = a_i \oplus b_i, 0 \leq i \leq 7$ 이 된다. 예를 들면, 다음의 표현들은 동치이다.

$$\begin{aligned} (x^6 + x^2 + x + 1) + (x^7 + x + 1) &= x^7 + x^6 + x^2 \quad (\text{다항식 표현}); \\ \{01000111\} \oplus \{10000011\} &= \{11000100\} \quad (\text{이진 표현}); \\ \{47\} \oplus \{83\} &= \{c4\} \quad (16\text{진법 표현}); \end{aligned}$$

4.2 곱셈 연산

블록 암호 ARIA에서 바이트들의 곱셈은 S-box 생성에 이용된다. ARIA 구현 시 S-box를 표 참조로 구현하는 경우에는 이 절의 내용이 필요하지 않다.

바이트들의 곱셈은 $GF(2^8) = \mathbb{Z}_2[x]/m(x)$ 에서의 곱셈으로 정의되며 \bullet 으로 표기한다. $m(x)$ 는 8차 기약 다항식으로 $m(x)$ 의 선택에 따라 $GF(2^8)$ 에서의 곱셈이 달라진다. 블록 암호 ARIA에서는 다음과 같은 기약다항식을 $GF(2^8)$ 정의에 사용한다.

$$m(x) = x^8 + x^4 + x^3 + x + 1.$$

$GF(2^8) = \mathbb{Z}_2[x]/m(x)$ 에서 두 다항식의 곱은 $\mathbb{Z}_2[x]$ 에서의 다항식 곱을 $m(x)$ 로 나눈 나머지로 주어진다.

예를 들면 $\{57\} \bullet \{83\} = \{c1\}$ 이 되는데 자세히 설명하면 다음과 같다.

$$\begin{aligned} (x^6 + x^4 + x^2 + x + 1)(x^7 + x + 1) &= x^{13} + x^{11} + x^9 + x^8 + x^7 + \\ &\quad x^7 + x^5 + x^3 + x^2 + x + \\ &\quad x^6 + x^4 + x^2 + x + 1 \\ &= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \end{aligned}$$

그리고,

$$\begin{aligned} x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \\ = (x^8 + x^4 + x^3 + x + 1)(x^5 + x^3) + x^7 + x^6 + 1 \end{aligned}$$

이므로 $(x^6 + x^4 + x^2 + x + 1) \bullet (x^7 + x + 1) = x^7 + x^6 + 1$ 이 되고 $x^7 + x^6 + 1$ 에 해당하는 16진법 표기는 $\{c1\}$ 이다.

위 곱셈은 결합 법칙을 만족하고 $\{01\}$ 은 곱셈에 대한 항등원이 된다. 0이 아닌 8차 미만의 이진 다항식 $b(x)$ 는 위 항등원에 대한 역원을 $b(x)^{-1}$ 로 표기한다.

$$b(x) \bullet b(x)^{-1} = 1.$$

역원은 Euclidean 알고리즘[4]을 이용하여 구할 수 있다.

5 알고리즘 구조

ARIA 알고리즘의 구조는 다음과 같다.

- 기본 구조: ISPN(Involutorial SPN) 구조
- 입·출력 크기: 128 비트
- 키 크기: 128,192,256 비트
- 라운드 키 크기: 128 비트
- 라운드 수: 키 크기에 따라 12,14,16 라운드

이 사양을 블록 단위(8 비트)로 정리하면 <표 3>과 같다. 이때, 입·출력 블록 크기를 Nb , 입력 키 블록 크기를 Nk , 그리고 라운드 수를 Nr 로 나타내었다. ARIA의 라운드

<표 3> ARIA 사양			
구 분	Nb	Nk	Nr
ARIA-128	16	16	12
ARIA-192	16	24	14
ARIA-256	16	32	16

함수는 다음과 같은 세 부분으로 구성되어 있다.

1. 라운드 키 덧셈(AddRoundKey): 128 비트 라운드 키를 라운드 입력 128 비트와 비트별 XOR한다.
2. 치환 계층(SubstLayer): 두 유형의 치환 계층이 있으며 각각은 2종의 8 비트 입·출력 S-box와 그들의 역변환으로 구성된다.
3. 확산 계층(DiffLayer): 간단한 16×16 involution 이진 행렬을 사용한 바이트 간의 확산 함수로 구성되어 있다.

본 알고리즘의 pseudo code는 (그림 1)과 같다. 각 부분 함수들은 이후 구체적으로 설명한다. 이때, $w[]$ 는 키 확장에 의해 생성된 라운드 키를 의미하며, 그 생성 과정은 5.3절에서 소개한다. 마지막 라운드는 확산 계층이 라운드 키 덧셈으로 대체된다. 암호화 과정과 복호화 과정은 (그림 2)와 같이 주어지며 암호화 과정과 복호화 과정은 라운드 키를 제외하고는 일치한다.

(그림 3)은 암호화 과정과 키 확장 과정을 엮어 평문에 대한 암호문을 출력하는 과정을 설명한 그림이다.

다음에서는 본 알고리즘을 구성하고 있는 각각의 계층들을 구체적으로 설명한다.

```

        Cipher(byte in[Nb], byte out[Nb], byte w[Nb*(Nr+1)])
begin
    byte state[Nb]

    state = in

    AddRoundKey(state, w[0..Nb-1])                // See Sec. 5.3

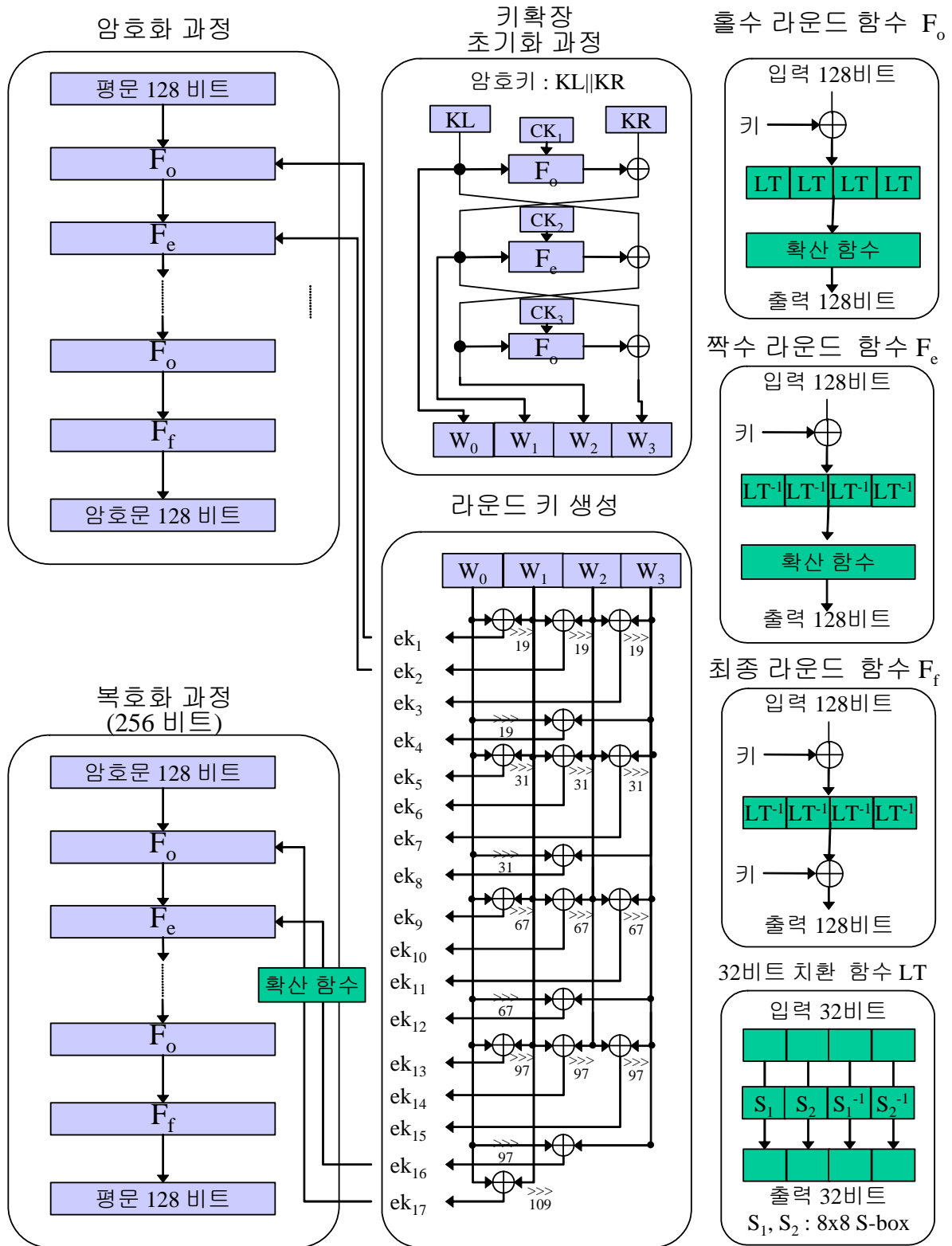
    for round = 1 to Nr-1
        SubstLayer(state)                          // See Sec. 5.1
        DiffLayer(state)                          // See Sec. 5.2
        AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
    end for

    SubstLayer(state)
    AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])

    out = state
end

```

(그림 1) ARIA의 Pseudo Code



(그림 2) 암호화 및 복호화 과정

5.1 치환 계층

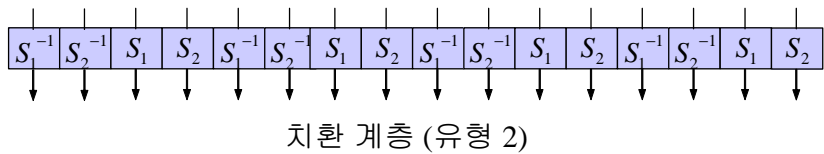
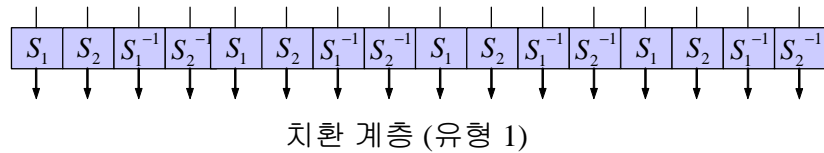
치환 계층은 8 비트 입·출력 S-box들로 구성된다. S-box들은 다음의 성질을 만족하도록 선택되었다.

- 최대 차분/선형 확률: 2^{-6}
- 대수적 차수: 7
- 고정점, 반고정점이 없을 것

이러한 성질을 만족하는 것들로는 유한체 $GF(2^8)$ 상의 함수 x^{-1} 에 아핀 변환을 취한 형태가 널리 사용되고 있다[6, 1]. x^{-1} 와 특성이 같은 것으로 $x^{-2^n}, n = 0, \dots, 7$ 이 있는데 ARIA에서는 x^{-1} 와 $x^{247}(=x^{-8})$ 에 아핀 변환을 취하여 S-box를 생성하였다. 즉, S-box S_1, S_2 는 다음의 꼴을 가진다.

$$S_1(x) = Bx^{-1} \oplus b, \quad S_2(x) = Cx^{247} \oplus c.$$

여기서, B, C 는 8×8 정칙 행렬(non-singular matrix)이며 b, c 는 8×1 행렬이다. ARIA에는 위에서 생성한 S_1, S_2 와 그 역치환 S_1^{-1}, S_2^{-1} 가 사용되며 구체적인 값은 <표 4>, <표 5>, <표 6>, <표 7>에 기술하였다. S-box의 값들은 모두 16진법으로 표현하였는데, S-box에 입력되는 8 비트 값이 16진법으로 'xy'이면, 출력은 'x'행 'y'열에 위치한 값이 된다. 예를 들면, $S_1(0x00) = 0x63$, $S_1(0x05) = 0x6b$, $S_1(0x72) = 0x40$ 과 같다. ARIA의 치



(그림 4) 두 유형의 치환 계층

환 계층은 두 유형(유형1, 유형2)으로 되어있는데 다음을 만족하도록 구성되었다.

- 두 유형 모두 S-box $S_1, S_2, S_1^{-1}, S_2^{-1}$ 로 구성
- 두 유형의 치환 계층은 서로 역의 관계 ($((\text{유형1})^{-1} = \text{유형2})$)

- 32-비트 단위로 4종의 S-box 사용

이러한 성질을 만족하는 두 유형의 치환 계층은 다음과 같다.

두 유형의 치환 계층은 (그림 2)와 같이 교대로 사용되어 involution 구조인 확산 계층과 함께 전체 구조가 involution 구조가 되도록 한다.

5.2 확산 계층

확산 계층은 ARIA와 다른 블록 암호를 구별 짓는 주요 부분으로 16×16 involution 이진 행렬을 사용한다. 확산 함수는 입력 16 바이트에 대하여 바이트 단위의 행렬 곱을 수행한 결과의 16 바이트를 출력으로 한다.

확산 계층 설계에 있어서 중요하게 고려되어야 하는 것은 안전성과 각종 적용 환경에서의 효율적 구현 가능성이다. ARIA의 확산 계층은 다음과 같은 설계 방향을 가지고 설계되었다.

- AES[6]에 강력한 분석 방법들에 대하여 내성을 가져야 한다. 이를 위해서 AES의 확산 함수 처리 단위(32 비트) 보다 큰 단위의 확산 함수를 사용한다.
- 8 비트, 32 비트 소프트웨어 및 하드웨어 구현에 적합해야 한다.
- 동종의 확산 함수 중에서 안전성과 효율성을 고려할 때 제일 우수해야 한다.

위 설계 방향을 만족하도록 ARIA의 확산 함수는 다음과 같이 선택되었다. ARIA의 확산 함수 $A: \text{GF}(2^8)^{16} \rightarrow \text{GF}(2^8)^{16}$ 는 입력을 $(x_0, x_1, \dots, x_{15})$ 라 하고 출력을 $(y_0, y_1, \dots, y_{15})$ 라 하면, 식 (5.1)에서와 같은 16×16 이진행렬의 곱으로 표현된다.

이때의 이진 행렬의 성질은 다음과 같다.

- A 는 involution 구조이다. 즉, $A^{-1} = A$ 이다.
- A 의 가지수(branch number)는 8이다. 여기서, 가지수 $\beta(A)$ 는 다음과 같이 정의된다.

$$\beta(A) = \min\{wt(x) + wt(A \cdot x) \mid x \in \text{GF}(2^8)^{16}, x \neq 0\},$$

$wt(x)$ 는 x 의 Hamming weight ²이다.

² x 에 포함된 '0'이 아닌 바이트 수

<표 4> S-box S_1

	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

<표 5> S-box S_1^{-1}

	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

<표 6> S-box S_2

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	e2	4e	54	fc	94	c2	4a	cc	62	0d	6a	46	3c	4d	8b	d1
1	5e	fa	64	cb	b4	97	be	2b	bc	77	2e	03	d3	19	59	c1
2	1d	06	41	6b	55	f0	99	69	ea	9c	18	ae	63	df	e7	bb
3	00	73	66	fb	96	4c	85	e4	3a	09	45	aa	0f	ee	10	eb
4	2d	7f	f4	29	ac	cf	ad	91	8d	78	c8	95	f9	2f	ce	cd
5	08	7a	88	38	5c	83	2a	28	47	db	b8	c7	93	a4	12	53
6	ff	87	0e	31	36	21	58	48	01	8e	37	74	32	ca	e9	b1
7	b7	ab	0c	d7	c4	56	42	26	07	98	60	d9	b6	b9	11	40
8	ec	20	8c	bd	a0	c9	84	04	49	23	f1	4f	50	1f	13	dc
9	d8	c0	9e	57	e3	c3	7b	65	3b	02	8f	3e	e8	25	92	e5
a	15	dd	fd	17	a9	bf	d4	9a	7e	c5	39	67	fe	76	9d	43
b	a7	e1	d0	f5	68	f2	1b	34	70	05	a3	8a	d5	79	86	a8
c	30	c6	51	4b	1e	a6	27	f6	35	d2	6e	24	16	82	5f	da
d	e6	75	a2	ef	2c	b2	1c	9f	5d	6f	80	0a	72	44	9b	6c
e	90	0b	5b	33	7d	5a	52	f3	61	a1	f7	b0	d6	3f	7c	6d
f	ed	14	e0	a5	3d	22	b3	f8	89	de	71	1a	af	ba	b5	81

<표 7> S-box S_2^{-1}

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	30	68	99	1b	87	b9	21	78	50	39	db	e1	72	09	62	3c
1	3e	7e	5e	8e	f1	a0	cc	a3	2a	1d	fb	b6	d6	20	c4	8d
2	81	65	f5	89	cb	9d	77	c6	57	43	56	17	d4	40	1a	4d
3	c0	63	6c	e3	b7	c8	64	6a	53	aa	38	98	0c	f4	9b	ed
4	7f	22	76	af	dd	3a	0b	58	67	88	06	c3	35	0d	01	8b
5	8c	c2	e6	5f	02	24	75	93	66	1e	e5	e2	54	d8	10	ce
6	7a	e8	08	2c	12	97	32	ab	b4	27	0a	23	df	ef	ca	d9
7	b8	fa	dc	31	6b	d1	ad	19	49	bd	51	96	ee	e4	a8	41
8	da	ff	cd	55	86	36	be	61	52	f8	bb	0e	82	48	69	9a
9	e0	47	9e	5c	04	4b	34	15	79	26	a7	de	29	ae	92	d7
a	84	e9	d2	ba	5d	f3	c5	b0	bf	a4	3b	71	44	46	2b	fc
b	eb	6f	d5	f6	14	fe	7c	70	5a	7d	fd	2f	18	83	16	a5
c	91	1f	05	95	74	a9	c1	5b	4a	85	6d	13	07	4f	4e	45
d	b2	0f	c9	1c	a6	bc	ec	73	90	7b	cf	59	8f	a1	f9	2d
e	f2	b1	00	94	37	9f	d0	2e	9c	6e	28	3f	80	f0	3d	d3
f	25	8a	b5	e7	42	b3	c7	ea	f7	4c	11	33	03	a2	ac	60

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{pmatrix} \quad (5.1)$$

5.3 키 확장

ARIA의 키 확장은 초기화 과정과 라운드 키 생성 과정의 두 부분으로 나뉜다.

5.3.1 초기화 과정

초기화 과정에서는 암/복호화 한 라운드를 F 함수로 하는 256 비트 입·출력 3라운드 Feistel 암호를 이용하여, 암호키 MK로부터 네 개의 128-비트 값 W_0, W_1, W_2, W_3 를 생성한다.

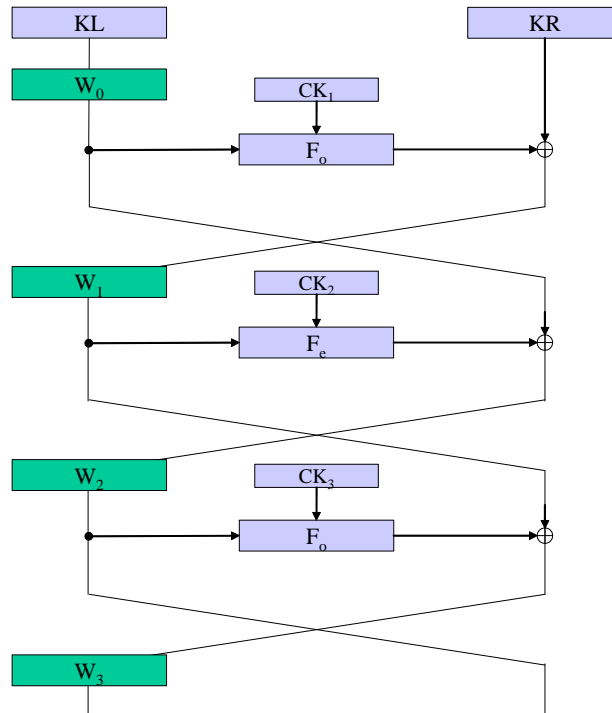
암호키 MK의 길이는 128, 192, 또는 256이므로 위 Feistel 암호의 입력에 필요한 256 비트 (KL, KR)을 다음과 같이 구성한다.

- 128 비트 KL는 MK의 상위 128 비트를 취한다.
- MK의 남은 비트를 이용하여 KR의 상위 비트를 채우고 나머지는 0으로 채운다.

$$KL || KR = MK || 0 \dots 0.$$

F_o 와 F_e 를 각각 홀수(치환 계층(유형1) 사용), 짝수(치환 계층(유형2) 사용) 라운드 함수라고 할 때 다음과 같이 W_0, W_1, W_2, W_3 를 생성한다.

$$\begin{aligned} W_0 &= KL, & W_2 &= F_e(W_1, CK_2) \oplus W_0, \\ W_1 &= F_o(W_0, CK_1) \oplus KR, & W_3 &= F_o(W_2, CK_3) \oplus W_1. \end{aligned}$$



(그림 5) 키 확장 초기화 과정

Feistel 암호의 128 비트 라운드 키 CK_i 는 π^{-1} 의 유리수 부분의 128 비트 상수

$$C1 = 0x517cc1b727220a94fe13abe8fa9a6ee0$$

$$C2 = 0x6db14acc9e21c820ff28b1d5ef5de2b0$$

$$C3 = 0xdb92371d2126e9700324977504e8c90e$$

를 이용하여 암호키의 길이에 따라 다음과 같이 결정된다.

<표 8> 암호키 길이에 따른 초기화 상수			
암호키 길이	CK_1	CK_2	CK_3
128-비트	$C1$	$C2$	$C3$
192-비트	$C2$	$C3$	$C1$
256-비트	$C3$	$C1$	$C2$

초기화 부분은 (그림 5)와 같다.

```

Keyexpansion(byte key[Nk], byte w[Nb*(Nr+1)])
begin

    KL = key[0..15]           // initial part
    KR = key[16..31]
    W[0] = KL
    W[1] =  $F_o(W[0], CK_1)$  XOR KR
    W[2] =  $F_e(W[1], CK_2)$  XOR W[0]
    W[3] =  $F_o(W[2], CK_3)$  XOR W[1]

    for i= 1 to Nr+1          // round key generation part
        w[i] = ROT(W[ ]) XOR ROT(W[ ])
    end for
end

```

(그림 6) ARIA 키 확장의 Pseudo Code

5.3.2 라운드 키 생성 과정

라운드 키 생성 과정에서는 네 개의 128 비트 W_0, W_1, W_2, W_3 를 조합하여 암호화 라운드 키 ek_i 와 복호화 라운드 키 dk_i 를 생성한다. 라운드 수는 암호키의 크기가 128, 192, 256 비트인 경우 각각 12, 14, 16 라운드이고 마지막 라운드에는 키 덧셈 계층이 두 번 있으므로 각각 13, 15, 17개의 라운드 키를 생성해야 한다. 암호화 라운드 키는 다음과 같다.

$$\begin{aligned}
 ek_1 &= (W_0) \oplus (W_1 \ggg^{19}), & ek_2 &= (W_1) \oplus (W_2 \ggg^{19}), \\
 ek_3 &= (W_2) \oplus (W_3 \ggg^{19}), & ek_4 &= (W_0 \ggg^{19}) \oplus (W_3), \\
 ek_5 &= (W_0) \oplus (W_1 \ggg^{31}), & ek_6 &= (W_1) \oplus (W_2 \ggg^{31}), \\
 ek_7 &= (W_2) \oplus (W_3 \ggg^{31}), & ek_8 &= (W_0 \ggg^{31}) \oplus (W_3), \\
 ek_9 &= (W_0) \oplus (W_1 \lll^{61}), & ek_{10} &= (W_1) \oplus (W_2 \lll^{61}), \\
 ek_{11} &= (W_2) \oplus (W_3 \lll^{61}), & ek_{12} &= (W_0 \lll^{61}) \oplus (W_3), \\
 ek_{13} &= (W_0) \oplus (W_1 \lll^{31}), & ek_{14} &= (W_1) \oplus (W_2 \lll^{31}), \\
 ek_{15} &= (W_2) \oplus (W_3 \lll^{31}), & ek_{16} &= (W_0 \lll^{31}) \oplus (W_3), \\
 ek_{17} &= (W_0) \oplus (W_1 \lll^{19})
 \end{aligned}$$

위에서 설명한 키 확장 과정을 정리하면 (그림 6)의 pseudo code로 나타낼 수 있다. 여기서 ROT()는 각 라운드별로 주어진 순환량에 따른 좌, 우 순환이동을 의미한다.

복호화 라운드 키는 암호화 라운드 키와 다르며 암호화 라운드 키로부터 유도된다. 먼저 키의 순서가 바뀌고 처음과 마지막 라운드 키를 제외하고 암호키를 입력으로 하는 확산 함수 A 의 출력이 복호화 라운드 키가 된다. 라운드 수가 n 일 때, 복호화 라운드 키는 다음과 같다.

$$dk_1 = ek_{n+1}, dk_2 = A(ek_n), dk_3 = A(ek_{n-1}), \dots, dk_n = A(ek_2), dk_{n+1} = ek_1.$$

6 암/복호화 예제

본 절에서는 위에서 서술한 ARIA 알고리즘에 의한 키 확장 및 평문 암/복호화 과정에서 중간값 변화를 예시하도록 한다. 예시는 키 크기 128/192/256에 대해 각각 제시되며, 중간값 상태를 나타내기 위해 다음과 같은 기호가 사용된다.

- input: 평문 입력값
- start: 라운드가 시작할 때의 라운드 입력값
- key_add: 입력과 키와의 XOR 후의 값
- s_box: 치환 계층 통과 후의 값
- diff_layer: 확산 계층 통과 후의 값
- output: 암호문 출력값

6.1 키 확장 : 128 비트 암호키

본 소절에서는 다음의 128 비트 키에 대한 키 확장 결과를 제시한다.

암호키 : 000102030405060708090a0b0c0d0e0f

키 확장에 사용되는 네 개의 값 W_0, W_1, W_2, W_3 을 <표 9>에 정리하였다.

<표 9> 키 확장을 위한 중간값: 128 비트

상수	값
W_0	000102030405060708090a0b0c0d0e0f
W_1	2afbea741e1746dd55c63ba1afcea0a5
W_2	7c8578018bb127e02dfe4e78c288e33c
W_3	6785b52b74da46bf181054082763ff6d

<표 10>은 위의 W_0, W_1, W_2, W_3 의 계산에 사용된 Feistel 구조의 중간값을 나타낸다.

암/복호화 라운드 키들을 <표 11>에 정리하였다.

<표 10> 키 확장 초기화: 128 비트

라운드	함수	값
1	key_round[1].input	000102030405060708090a0b0c0d0e0f
	key_round[1].key_add	517dc3b423270c93f61aa1e3f69760ef
	key_round[1].s_box	d1b933142669815c422ef194426590d3
	key_round[1].diff_layer	2afbea741e1746dd55c63ba1afcea0a5
2	key_round[2].start	2afbea741e1746dd55c63ba1afcea0a5
	key_round[2].key_add	474aa0b880368efdaaee8a7440934215
	key_round[2].s_box	1606e0703a6419ba623d7ec4725c2c97
	key_round[2].diff_layer	7c847a028fb421e725f74473ce85ed33
3	key_round[3].start	7c8578018bb127e02dfe4e78c288e33c
	key_round[3].key_add	a7174f1caa97ce902edad90dc6602a32
	key_round[3].s_box	5c2b92d6ac65ece03180e509b4ff956c
	key_round[3].diff_layer	4d7e5f5f6acd00624dd66fa988ad5fc8
	key_round[3].output	6785b52b74da46bf181054082763ff6d

<표 11> 키 생성 — 128 비트

라운드	암호화 키	복호화 키
1	d415a75c794b85c5e0d2a0b3cb793bf6	0f0aa16daee61bd7dfce5a599970fb35
2	369c65e4b11777ab713a3e1e6601b8f4	ccb3a0230b6dac1d53eef49d961aa57f
3	0368d4f13d14497b6529ad7ac809e7d0	60ea3252ac3ea9bc9ac78e79df20b5b5
4	c644552b549a263fb8d0b50906229eec	5794eadaece652f8a2ccbf68ee82a730
5	5f9c434951f2d2ef342787b1a781794c	468a335e49ec1db45d112aaf2109e5bf
6	afea2c0ce71db6de42a47461f4323c54	938ebbda880c6bb87fa01c97e68811a9
7	324286db44ba4db6c44ac306f2a84b2c	bfd5018ab33d14cc538ea5c81bd1011
8	7f9fa93574d842b9101a58063771eb7b	b5a90e77d5b94bb56e47af759fcfa05e
9	aab9c57731fcd213ad5677458fcfe6d4	21a6c28c5e1175a4378cd34dd3195a83
10	2f4423bb06465abada5694a19eb88459	8d726063ca2ceddc92afb45dd7db643e
11	9f8772808f5d580d810ef8ddac13abeb	27efd355eb17e90e5963c46515016f8d
12	8684946a155be77ef810744847e35fad	d000e81367819b077b0a657f6740e8e4
	0f0aa16daee61bd7dfce5a599970fb35	d415a75c794b85c5e0d2a0b3cb793bf6

6.2 키 확장 : 192 비트 암호키

본 소절에서는 192 비트 키에 대한 키 확장 결과를 제시한다. 내용의 구성은 전 소절의 128 비트의 경우와 동일하므로 설명은 생략하도록 한다.

암호키 : 000102030405060708090a0b0c0d0e0f1011121314151617

<표 12> 키 확장을 위한 중간값: 192 비트

상수	값
W_0	000102030405060708090a0b0c0d0e0f
W_1	e48c52301e91d991b649ed7bb7cde8ad
W_2	a356ea6cafe4869797a1b4eea56d38cc
W_3	e1898f2e0e626ccf1f58bd50713c93bb

<표 13> 키 확장 초기화: 192 비트

라운드	함수	값
1	key_round[1].input	000102030405060708090a0b0c0d0e0f
	key_round[1].key_add	6db048cf9a24ce27f721bbdee350ecbf
	key_round[1].s_box	3ca7d445b855ecc66806fef9110883a5
	key_round[1].diff_layer	f49d40230a84cf86b649ed7bb7cde8ad
2	key_round[2].start	e48c52301e91d991b649ed7bb7cde8ad
	key_round[2].key_add	3f1e652d3fb730e1b56d7a0eb32521a3
	key_round[2].s_box	25c44ddf2570040bd2efda8b4b9dfd17
	key_round[2].diff_layer	a357e86fabe180909fa8bee5a96036c3
3	key_round[3].start	a356ea6cafe4869797a1b4eea56d38cc
	key_round[3].key_add	f22a2bdb88c68c0369b21f065ff7562c
	key_round[3].s_box	89180b59c427f01bf9d0cb21cff8b9d4
	key_round[3].diff_layer	0505dd1e10f3b55ea911502bc6f17b16
	key_round[3].output	e1898f2e0e626ccf1f58bd50713c93bb

<표 14> 키 생성 — 192 비트

라운드	암호화 키	복호화 키
1	bd14be928e4305d5333b3cc231a278f6	a467dc0b2048d83faf3ffd3355a9ff5b
2	4395c65ac3dc4c6d269b1f8f81503c00	4dd0b9831c584a7f72e931dd8ede23f5
3	3121965d9e01475bda385705b2c736eb	b4a02c5b7e7cff4981137b76a1e8af63
4	40486f2e2e220c4fbf985c51507df23a	e58ecdefea05c868b394d7dabc7298b3
5	6f9ad358cd1da267352ab928609ed4f8	dc94b739beef8d4acd30f3fb4bbd0a19
6	ae5623a9583c0d48e980e054988e8170	518c97ed5d0bfaac5240e7f2dd6e2087
7	412fcd1b6cf798cb8b656d709bdc426c	a3cffe0ff0406d3f6fcce825184d8f470
8	f99393300e6068c91752b15e612e87ad	bab3d61669640fcd9fda3fd0cce02c65
9	36c83fac72fcbb12b498804d0fdf353d	9fe831feae7e081014a882cd3f3396f3
10	167864adca3c7e88222330362231787f	9ad65f61fd36b359f684ac037cc53668
11	40bdfdc6a1c314e0eb90850b64a17555	622efcce90b0b68380d667bfeee031ed
12	0088ae6f6fe3cd0eff589d1011bc337b	a44f0daf2b4489a972d5971afdb359f2
13	0f49eecbdf21f0bad3effe5dfe4b2717	9777f3c8a450e691aed77bb2243af84e
14	b37e117bd54103e6e4ff711de6669d9b	044b0b0eccb4bdfb37747a14e7512e75
	a467dc0b2048d83faf3ffd3355a9ff5b	bd14be928e4305d5333b3cc231a278f6

6.3 키 확장 : 256 비트 암호키

본 소절에서는 256 비트 키에 대한 키 확장 결과를 제시한다. 내용의 구성은 전 소절들의 128, 192 비트의 경우와 동일하므로 설명은 생략하도록 한다.

암호키 : 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

<표 15> 키 확장을 위한 중간값: 256 비트

상수	값
W_0	000102030405060708090a0b0c0d0e0f
W_1	15169e6ec54aaf0c975414fead1c71f3
W_2	90ec92c2a800405af99389e9c88b4e62
W_3	b68cd7a1ba16abee905f7009a8e968a9

<표 16> 키 확장 초기화: 256 비트

라운드	함수	값
1	key_round[1].input	000102030405060708090a0b0c0d0e0f
	key_round[1].key_add	db93351e2523ef770b2d9d7e08e5c701
	key_round[1].s_box	b957d9c43f6b61192bdf75a8305a3168
	key_round[1].diff_layer	05078c7dd15fb91b8f4d0ee5b1016fec
2	key_round[2].start	15169e6ec54aaf0c975414fead1c71f3
	key_round[2].key_add	446a5fd9e268a5986947bf1657861f13
	key_round[2].s_box	860acf6f3bb4063be45808bedabec0cb
	key_round[2].diff_layer	90ed90c1ac05465df19a83e2c486406d
3	key_round[3].start	90ec92c2a800405af99389e9c88b4e62
	key_round[3].key_add	fd5dd80e3621887a06bb383c27d6acd2
	key_round[3].s_box	54a42d62050697516f8a760ccc1caac9
	key_round[3].diff_layer	a39a49cf7f5c04e2070b64f705f5195a
	key_round[3].output	b68cd7a1ba16abee905f7009a8e968a9

<표 17> 키 생성 — 256 비트

라운드	암호화 키	복호화 키
1	8e3f60a1d7c8deae5de898e18e92dbac	f37728567c61bca7affc62e88395a6bb
2	7cdacc735712fa0c9f5f4bccdc2148e2	fd62e7342bfde4dbd9499dbb605b04bf
3	bdf9a41332f477182cee5be2268a7b7f	0fddde54a27fd06456f6779dcf03c84d
4	174d37a19a56cb6e309f910889a80928	a1d0cf146ae42a0d5dad9e70003bdec0
5	5a39e1e52e283ada829c541222a527f2	d8e599b24da2fa817d5093a776793345
6	840002abe4938a89c754944b5e3b6220	bfc0e457ae1820ff7e0efbbd80db257e
7	c13e4391c519ef198dbede34e835ae71	ec43f94c756c16adb2acd64b19b6c5d8
8	ae96cbbfba14afe898557c07b8fb7cbf	80f83e941cc54e3630d40048da4028d7
9	92eb809cd1a688396aabd9c6d4a45bee	fbe497edad612923021e12d063a84f41
10	4a24ef53fc5bc6c0c54986a6f81c79f8	e0fdffae2d79e75a524013b7f04991a8
11	42e77cc39d1d6d4fcf42131dffcf99b1f	13e967b0fdd52624b97bedf6e9c0a883
12	578df6e0db970a2f705f5049c869c869	0ff5cf18dbcdaac866ea6cac40442506
13	62a455854faf0c785e8732f286864138	6dddbb6cd6ef2ef19466c369942c10fa
14	4116be43b9836bf87311b3cfe56a3892	fe0124176f59613ebf4fbb7dc11ae43f
15	4de7c735e02ff85e2de73dbd13cd25b2	65d35c19276918ffc078ed2e183c1f93
16	348e54a23e122eeb1659f70e28e9e9a8	772cce8c5b4055fd75160b2faf0c9165
	f37728567c61bca7affc62e88395a6bb	8e3f60a1d7c8deae5de898e18e92dbac

6.4 암호/복호화 예제 : 128 비트 암호키

본 소절에서는 128 비트 키에 대한 암호/복호화 결과의 중간값들을 제시하도록 한다. 평문은 본 소절 뿐 아니라 그 뒤 소절의 192, 256 비트의 경우에 대해서도 공통적으로

00112233445566778899aabbccddeeff

를 사용하기로 한다. 또한 128 비트의 경우 키로는

000102030405060708090a0b0c0d0e0f

를 사용한 결과를 제시한다.

<표 18> ARIA-128 암호화 예제 벡터

라운드	함수	값
1	round[0].input	00112233445566778899aabbccddeeff
	round[1].start	00112233445566778899aabbccddeeff
	round[1].key_add	d404856f3d1ee3b2684b0a0807a4d509
	round[1].s_box	489467d927594dd54595a350c5a9b539
	round[1].diff_lay	7fc7f12befd0a0791de87fa96b469f52
2	round[2].start	7fc7f12befd0a0791de87fa96b469f52
	round[2].key_add	495b94cf5ec7d7d26cd241b70d4727a6
	round[2].s_box	a4e222da9d5b0ea2b8c98334f358ccd4
	round[2].diff_lay	ac8de17e49f7c5117618993162b189e9
3	round[3].start	ac8de17e49f7c5117618993162b189e9
	round[3].key_add	afe5358f74e38c6a1331344baab86e39
	round[3].s_box	795ad99a9233f00a7d7328c3ac7045aa
	round[3].diff_lay	c3e8d59ec2e62d5249ca2741653cb7dd
4	round[4].start	c3e8d59ec2e62d5249ca2741653cb7dd
	round[4].key_add	05ac80b5967c0b6df11a9248631e2931
	round[4].s_box	3644cdf235ee2bca2bfb4f8d00c4a573
	round[4].diff_lay	5d4aebb165e141ff759f669e1e85cc45
5	round[5].start	5d4aebb165e141ff759f669e1e85cc45
	round[5].key_add	02d6a8f83413931041b8e12fb904b509
	round[5].s_box	771c6ff718cb223e8370e04d5694d239
	round[5].diff_lay	7806e469f68874c5004b5f4a046bbcfa
6	round[6].start	7806e469f68874c5004b5f4a046bbcfa
	round[6].key_add	d7ecc8651195c21b42ef2b2bf05980ae
뒷 페이지에서 계속		

전 페이지 이어 계속		
라운드	함수	값
	round[6].s_box	0d80e821e34b2503f6d3f1ae171ecd9d
	round[6].diff_lay	110f93c9a630cdd51f97d2202413345a
7	round[7].start	110f93c9a630cdd51f97d2202413345a
	round[7].key_add	234d1512e28a8063dbdd1126d6bb7f76
	round[7].s_box	262f2f5e98f13a2cb944e377f68a6bad
	round[7].diff_lay	e054428ef088fef97928241cd3be499e
8	round[8].start	e054428ef088fef97928241cd3be499e
	round[8].key_add	9fcbebbb8450bc4069327c1ae4cfa2e5
	round[8].s_box	6e13e98a4f8c652de46c102eae453a5a
	round[8].diff_lay	5734f38ea1ca3ddd102e71f95e1d5f97
9	round[9].start	5734f38ea1ca3ddd102e71f95e1d5f97
	round[9].key_add	fd8d36f99036efcebd7806bcd1d2b943
	round[9].s_box	541f244c6085614e7a07a5183ea2dbaf
	round[9].diff_lay	4903325be3e500cccd52fba4354a39ae
10	round[10].start	4903325be3e500cccd52fba4354a39ae
	round[10].key_add	664711e0e5a35a7617046f05abf2bdf7
	round[10].s_box	d35882902ababe428787a8c20eb57af8
	round[10].diff_lay	cb8c508e2c4f87880639dc896d25ec9d
11	round[11].start	cb8c508e2c4f87880639dc896d25ec9d
	round[11].key_add	540b220ea312df8587372454c1364776
	round[11].s_box	204694620a64ef3617e4a602788516ad
	round[11].diff_lay	e7e0d2457ed73d23d481424095afdca0
12	round[12].start	e7e0d2457ed73d23d481424095afdca0
	round[12].key_add	6164462f6b8cda5d2c913608d24c830d
	round[12].s_box	d8125abb058257a4424705627f35ec4d
	round[12].output	d718fbd6ab644c739da95f3be6451778

<표 19> ARIA-128 복호화 예제 벡터

라운드	함수	값
1	round[0].input	d718fbd6ab644c739da95f3be6451778
	round[1].start	d718fbd6ab644c739da95f3be6451778
	round[1].key_add	d8125abb058257a4424705627f35ec4d
	round[1].s_box	6164462f6b8cda5d2c913608d24c830d
	round[1].diff_lay	ecf534410109432b440a529fee9fb3d2
2	round[2].start	ecf534410109432b440a529fee9fb3d2
	round[2].key_add	204694620a64ef3617e4a602788516ad
	round[2].s_box	540b220ea312df8587372454c1364776
	round[2].diff_lay	b3b2b0c2868417fe1d4026bbd195cf4d
3	round[3].start	b3b2b0c2868417fe1d4026bbd195cf4d
	round[3].key_add	d35882902ababe428787a8c20eb57af8
	round[3].s_box	664711e0e5a35a7617046f05abf2bdf7
	round[3].diff_lay	038bce968c6333b6d8cb1a70d0207c9f
4	round[4].start	038bce968c6333b6d8cb1a70d0207c9f
	round[4].key_add	541f244c6085614e7a07a5183ea2dbaf
	round[4].s_box	fd8d36f99036efcebd7806bcd1d2b943
	round[4].diff_lay	2899dad406607899b97d3a818f4cdfe5
5	round[5].start	2899dad406607899b97d3a818f4cdfe5
	round[5].key_add	6e13e98a4f8c652de46c102eae453a5a
	round[5].s_box	9fcbebbb8450bc4069327c1ae4cfa2e5
	round[5].diff_lay	b5a1948410fd5194c6e4ffe010027a04
6	round[6].start	b5a1948410fd5194c6e4ffe010027a04
	round[6].key_add	262f2f5e98f13a2cb944e377f68a6bad
	round[6].s_box	234d1512e28a8063dbdd1126d6bb7f76
	round[6].diff_lay	b25ab8394878f44f33eb1bf296a3dd8c
7	round[7].start	b25ab8394878f44f33eb1bf296a3dd8c
	round[7].key_add	0d80e821e34b2503f6d3f1ae171ecd9d
	round[7].s_box	d7ecc8651195c21b42ef2b2bf05980ae
	round[7].diff_lay	c2b56180cd72698bed374f38c95b7267
8	round[8].start	c2b56180cd72698bed374f38c95b7267
	round[8].key_add	771c6ff718cb223e8370e04d5694d239
	round[8].s_box	02d6a8f83413931041b8e12fb904b509
	round[8].diff_lay	17e20f7e6bff5e6e1c779cc0d3ddfff0
뒷 페이지에서 계속		

전 페이지 이어 계속		
라운드	함수	값
9	round[9].start	17e20f7e6bff5e6e1c779cc0d3ddfff0
	round[9].key_add	3644cdf235ee2bca2bfb4f8d00c4a573
	round[9].s_box	05ac80b5967c0b6df11a9248631e2931
	round[9].diff_lay	f428b9f9581f1dd6efdc9c9e7bab2194
10	round[10].start	f428b9f9581f1dd6efdc9c9e7bab2194
	round[10].key_add	795ad99a9233f00a7d7328c3ac7045aa
	round[10].s_box	afe5358f74e38c6a1331344baab86e39
	round[10].diff_lay	830df18f764ce7ace1aa4751e659a359
11	round[11].start	830df18f764ce7ace1aa4751e659a359
	round[11].key_add	a4e222da9d5b0ea2b8c98334f358ccd4
	round[11].s_box	495b94cf5ec7d7d26cd241b70d4727a6
	round[11].diff_lay	98948fca40d8d6d23e9fc62fa2e95ddd
12	round[12].start	98948fca40d8d6d23e9fc62fa2e95ddd
	round[12].key_add	489467d927594dd54595a350c5a9b539
	round[12].s_box	d404856f3d1ee3b2684b0a0807a4d509
	round[12].output	00112233445566778899aabbccddeeff

6.5 암호/복호화 예제 : 192 비트 암호키

본 소절에서는 192 비트 키에 대한 암호/복호화 결과의 중간값들을 제시하도록 한다. 평문은 128 비트의 경우와 동일하며, 키는

000102030405060708090a0b0c0d0e0f1011121314151617

를 사용한 결과를 제시한다.

<표 20> ARIA-192 암호화 예제 벡터

라운드	함수	값
1	round[0].input	00112233445566778899aabbccddeeff
	round[1].start	00112233445566778899aabbccddeeff
	round[1].key_add	bd059ca1ca1663a2bba29679fd7f9609
	round[1].s_box	7ac21ce974be00d2eafd35bd54403539
	round[1].diff_lay	ff0a53eb839b686852dad8cf18de2cf2
2	round[2].start	ff0a53eb839b686852dad8cf18de2cf2
	round[2].key_add	bc9f95b1404724057441c740998e10f2
	round[2].s_box	78d72ae1725836c2ca22c62df969cae0
	round[2].diff_lay	ee4161aac78ae47750dfe66aff08763b
3	round[3].start	ee4161aac78ae47750dfe66aff08763b
	round[3].key_add	df60f7f7598ba32c8ae7b16f4dcf40d0
	round[3].s_box	9eff26eacb4f71d47ef356d9e3da72b2
	round[3].diff_lay	75619b2290bbf0fa4017e4b1b523a8c7
4	round[4].start	75619b2290bbf0fa4017e4b1b523a8c7
	round[4].key_add	3529f40cbe99fcb5ff8fb8e0e55e5afd
	round[4].s_box	d943bf3c5a26b0f27d9a6c902a10beba
	round[4].diff_lay	9f1cc55fa9d75818f408c2259ea62d2b
5	round[5].start	9f1cc55fa9d75818f408c2259ea62d2b
	round[5].key_add	f086160764cafa7fc1227b0dfe38f9d3
	round[5].s_box	8c84ff78436e144178410309bb3a691c
	round[5].diff_lay	454e7efa1988f1182fa9a316dde0f831
6	round[6].start	454e7efa1988f1182fa9a316dde0f831
	round[6].key_add	eb185d5341b4fc50c6294342456e7941
	round[6].s_box	3c2a4c38f814b008c7431af468cab67f
	round[6].diff_lay	88c39bb29e7a83334998a21901abe928
7	round[7].start	88c39bb29e7a83334998a21901abe928
뒷 페이지에서 계속		

전 페이지 이어 계속		
라운드	함수	값
	round[7].key_add	c9ec56a9f28d1bf8c2fdcf699a77ab44
	round[7].s_box	ddd6b9a4891f44f725ba5f27b8260edd
	round[7].diff_lay	deab0665aacde8aad1c1986ff9967c5e
8	round[8].start	deab0665aacde8aad1c1986ff9967c5e
	round[8].key_add	27389555a4ad8063c693293198b8fbf3
	round[8].s_box	3d532a831d46cd31c75ca573e25a0fa5
	round[8].diff_lay	9d8112c94f9e6711187b1a34487ac5e5
9	round[9].start	9d8112c94f9e6711187b1a34487ac5e5
	round[9].key_add	ab492d653d62dc03ace39a7947a5f0d8
	round[9].s_box	6278fa97270e931b913337bda0bf1790
	round[9].diff_lay	297d76553d4b984f3e8d4ed52464459d
10	round[10].start	297d76553d4b984f3e8d4ed52464459d
	round[10].key_add	3f0512f8f777e6c71cae7ee306553de2
	round[10].s_box	25b9c98926198ef6c42bf333a524275b
	round[10].diff_lay	cd372f097eb243c8c0ba6c39b5c7ac23
11	round[11].start	cd372f097eb243c8c0ba6c39b5c7ac23
	round[11].key_add	8d8ad2cfd7f157282b2ae932d166d976
	round[11].s_box	5df17f459eabda57f118eb6c3e58e5ad
	round[11].diff_lay	55f9a19b5c91e0957b6a4f3049d7bc0c
12	round[12].start	55f9a19b5c91e0957b6a4f3049d7bc0c
	round[12].key_add	55710ff433722d9b8432d220586b8f77
	round[12].s_box	edfa763d66dcd83e4f6cb51d5e237326
	round[12].diff_lay	f0cf94f7402d0130ff23055245ef68ea
13	round[13].start	f0cf94f7402d0130ff23055245ef68ea
	round[13].key_add	ff867a3c9f0cf18a2cccfb0fbba44ffd
	round[13].s_box	1684bd0cdb3c2bbb7116633ceaa992a2
	round[13].diff_lay	a01563f5ea1626ad9ac1dcbf6931456e
14	round[14].start	a01563f5ea1626ad9ac1dcbf6931456e
	round[14].key_add	136b728e3f57254b7e3eada28f57d8f5
	round[14].s_box	8223401325933f958a9b95fd73936122
	round[14].output	26449c1805dbe7aa25a468ce263a9e79

<표 21> ARIA-192 복호화 예제 벡터

라운드	함수	값
1	round[0].input	26449c1805dbe7aa25a468ce263a9e79
	round[1].start	26449c1805dbe7aa25a468ce263a9e79
	round[1].key_add	8223401325933f958a9b95fd73936122
	round[1].s_box	136b728e3f57254b7e3eada28f57d8f5
	round[1].diff_lay	5b54048fc76461c403ff52e16477b157
2	round[2].start	5b54048fc76461c403ff52e16477b157
	round[2].key_add	1684bd0cdb3c2bbb7116633ceaa992a2
	round[2].s_box	ff867a3c9f0cf18a2cccfb0fbba44ffd
	round[2].diff_lay	595a5a6618a02777ce7fce6bffcbdc45
3	round[3].start	595a5a6618a02777ce7fce6bffcbdc45
	round[3].key_add	edfa763d66dcd83e4f6cb51d5e237326
	round[3].s_box	55710ff433722d9b8432d220586b8f77
	round[3].diff_lay	b87fb2aa74ae123f428c3cb6822a7d1e
4	round[4].start	b87fb2aa74ae123f428c3cb6822a7d1e
	round[4].key_add	5df17f459eabda57f118eb6c3e58e5ad
	round[4].s_box	8d8ad2cfd7f157282b2ae932d166d976
	round[4].diff_lay	f92d7eb098f603bc091b00c8ee992d42
5	round[5].start	f92d7eb098f603bc091b00c8ee992d42
	round[5].key_add	25b9c98926198ef6c42bf333a524275b
	round[5].s_box	3f0512f8f777e6c71cae7ee306553de2
	round[5].diff_lay	33f46d7a7a0569b7c373d04f7dd13717
6	round[6].start	33f46d7a7a0569b7c373d04f7dd13717
	round[6].key_add	6278fa97270e931b913337bda0bf1790
	round[6].s_box	ab492d653d62dc03ace39a7947a5f0d8
	round[6].diff_lay	9e9cd47c19401ec73b9227226682fbd5
7	round[7].start	9e9cd47c19401ec73b9227226682fbd5
	round[7].key_add	3d532a831d46cd31c75ca573e25a0fa5
	round[7].s_box	27389555a4ad8063c693293198b8fbf3
	round[7].diff_lay	67656fb2e07b4b3aba6060f774c622b8
8	round[8].start	67656fb2e07b4b3aba6060f774c622b8
	round[8].key_add	ddd6b9a4891f44f725ba5f27b8260edd
	round[8].s_box	c9ec56a9f28d1bf8c2fdcf699a77ab44
	round[8].diff_lay	a3c27dc6566ab818d3eb983957f9208c
뒷 페이지에서 계속		

전 페이지 이어 계속		
라운드	함수	값
9	round[9].start	a3c27dc6566ab818d3eb983957f9208c
	round[9].key_add	3c2a4c38f814b008c7431af468cab67f
	round[9].s_box	eb185d5341b4fc50c6294342456e7941
	round[9].diff_lay	1652a019be58a7188ec5af0ac7ff5f74
10	round[10].start	1652a019be58a7188ec5af0ac7ff5f74
	round[10].key_add	8c84ff78436e144178410309bb3a691c
	round[10].s_box	f086160764cafa7fc1227b0dfe38f9d3
	round[10].diff_lay	bb6d43f2ca960671fd4c0b2fc4f08f57
11	round[11].start	bb6d43f2ca960671fd4c0b2fc4f08f57
	round[11].key_add	d943bf3c5a26b0f27d9a6c902a10beba
	round[11].s_box	3529f40cbe99fcb5ff8fb8e0e55e5afd
	round[11].diff_lay	3ab02b45e00bf87d0c26c1c31e692b40
12	round[12].start	3ab02b45e00bf87d0c26c1c31e692b40
	round[12].key_add	9eff26eacb4f71d47ef356d9e3da72b2
	round[12].s_box	df60f7f7598ba32c8ae7b16f4dcf40d0
	round[12].diff_lay	efa0d929d608d05364f5bd9fdd5332ae
13	round[13].start	efa0d929d608d05364f5bd9fdd5332ae
	round[13].key_add	78d72ae1725836c2ca22c62df969cae0
	round[13].s_box	bc9f95b1404724057441c740998e10f2
	round[13].diff_lay	7e8917e7b80abd29dd894fa9b3111b4c
14	round[14].start	7e8917e7b80abd29dd894fa9b3111b4c
	round[14].key_add	7ac21ce974be00d2eafd35bd54403539
	round[14].s_box	bd059ca1ca1663a2bba29679fd7f9609
	round[14].output	00112233445566778899aabbccddeeff

6.6 암호/복호화 예제 : 256 비트 암호키

본 소절에서는 256 비트 키에 대한 암호/복호화 결과의 중간값들을 제시하도록 한다. 평문은 128, 192 비트의 경우와 동일하며, 키는

000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f

를 사용한 결과를 제시한다.

<표 22> ARIA-256 암호화 예제 벡터

라운드	함수	값
1	round[0].input	00112233445566778899aabbccddeeff
	round[1].start	00112233445566778899aabbccddeeff
	round[1].key_add	8e2e4292939db8d9d571325a424f3553
	round[1].s_box	19e7f69edc259a7b03aba1e52ccdd95f
	round[1].diff_lay	64739617aa297fe46a5146919209e915
2	round[2].start	64739617aa297fe46a5146919209e915
	round[2].key_add	18a95a64fd3b85e8f50e0d5d4e28a1f7
	round[2].s_box	34a4be36219897617762d7a4b65732f8
	round[2].diff_lay	f01c2fdb0bccbf37a8bf5f2e9c034ffb
3	round[3].start	f01c2fdb0bccbf37a8bf5f2e9c034ffb
	round[3].key_add	4de58bc83938c82f845104ccba893484
	round[3].s_box	e35ace4a123ab14d5f7a3007f4232886
	round[3].diff_lay	c7eebca8e1e6fd2e73e9f57de119d455
4	round[4].start	c7eebca8e1e6fd2e73e9f57de119d455
	round[4].key_add	d0a38b097bb036404376647568b1dd7d
	round[4].s_box	60ba3d0d03eb052d64ad4356f76fc1b9
	round[4].diff_lay	6c7ce71dfc22061861546c85a30dbff1
5	round[5].start	6c7ce71dfc22061861546c85a30dbff1
	round[5].key_add	364506f8d20a3cc2e3c8389781a89803
	round[5].s_box	05cfa5f7b56a6d05113576150c7ee21b
	round[5].diff_lay	97f9639537379423693621392e83efc9
6	round[6].start	97f9639537379423693621392e83efc9
	round[6].key_add	13f9613ed3a41eaaae62b57270b88de9
	round[6].s_box	824cef10a95d7239be08d50cd05a5da1
	round[6].diff_lay	7a4c3f387ed40316706095ea3ce83f9d
7	round[7].start	7a4c3f387ed40316706095ea3ce83f9d
뒷 페이지에서 계속		

전 페이지 이어 계속		
라운드	함수	값
	round[7].key_add	bb727ca9bbcddec0ffdde4bded4dd91ec
	round[7].s_box	ea0c01a4ea82833c549bccf94844ac80
	round[7].diff_lay	eab89889e8398c8a38fa340c982de87d
8	round[8].start	eab89889e8398c8a38fa340c982de87d
	round[8].key_add	442e5336522d2362a0af480b20d694c2
	round[8].s_box	861aed854840260e47fc524654ec2251
	round[8].diff_lay	9e1d6512590a730035caf4a431d293bb
9	round[9].start	9e1d6512590a730035caf4a431d293bb
	round[9].key_add	0cf6e58e88acfb395f612d62e576c855
	round[9].s_box	feb32a69c4fe63aacf87fa08d942b124
	round[9].diff_lay	75cb1bab78f698e5bfb077c2062993b2
10	round[10].start	75cb1bab78f698e5bfb077c2062993b2
	round[10].key_add	3feff4f884ad5e257af9f164fe35ea4a
	round[10].s_box	25d3bf894f4658f0bd4ca1360cc887c8
	round[10].diff_lay	203c974b18b7434de855954eb2d058b1
11	round[11].start	203c974b18b7434de855954eb2d058b1
	round[11].key_add	62dbeb8885aa2e02271786534d19c3ae
	round[11].s_box	aa0a3c529739c399cc2bdc5fe377332b
	round[11].diff_lay	a5b315cd24206c9c2ed5049bfb511a3
12	round[12].start	a5b315cd24206c9c2ed5049bfb511a3
	round[12].key_add	f23ee32dfffb766b35e8a54d233acd9ca
	round[12].s_box	049b11df7d7033f59dbb20a26644356e
	round[12].diff_lay	c6ba5f7201f9596a1d2d3aae33e4fa54
13	round[13].start	c6ba5f7201f9596a1d2d3aae33e4fa54
	round[13].key_add	a41e0af74e56551243aa085cb562bb6c
	round[13].s_box	4959a3ea2f2aed5e1a39bf54d50efedf
	round[13].diff_lay	fbfe7a26af3be9cb0fa8452af1bb3585
14	round[14].start	fbfe7a26af3be9cb0fa8452af1bb3585
	round[14].key_add	bae8c46516b882337cb9f6e514d10d17
	round[14].s_box	c09c1c21ff5a13fb017d425a9b0fd72b
	round[14].diff_lay	6971d8a121818c613e035108d445b44d
15	round[15].start	6971d8a121818c613e035108d445b44d
	round[15].key_add	24961f94c1ae743f13e46cb5c78891ff
	round[15].s_box	367bcb04789dcaed7d7db8fec649ac60
뒷 페이지에서 계속		

전 페이지 이어 계속		
라운드	함수	값
	round[15].diff_lay	531d29e52f0e5ab9498ecc4dd299f8f0
16	round[16].start	531d29e52f0e5ab9498ecc4dd299f8f0
	round[16].key_add	67937d47111c74525fd73b43fa701158
	round[16].s_box	0a5cff91e3d692888473e22914b88247
	round[16].output	f92bd7c79fb72e2f2b8f80c1972d24fc

<표 23> ARIA-256 복호화 예제 벡터

라운드	함수	값
1	round[0].input	f92bd7c79fb72e2f2b8f80c1972d24fc
	round[1].start	f92bd7c79fb72e2f2b8f80c1972d24fc
	round[1].key_add	0a5cff91e3d692888473e22914b88247
	round[1].s_box	67937d47111c74525fd73b43fa701158
	round[1].diff_lay	cb192c3053602e36a4342545a612a8df
2	round[2].start	cb192c3053602e36a4342545a612a8df
	round[2].key_add	367bcb04789dcaed7d7db8fec649ac60
	round[2].s_box	24961f94c1ae743f13e46cb5c78891ff
	round[2].diff_lay	cf41c2755d25c39f578b35c7540c1f66
3	round[3].start	cf41c2755d25c39f578b35c7540c1f66
	round[3].key_add	c09c1c21ff5a13fb017d425a9b0fd72b
	round[3].s_box	bae8c46516b882337cb9f6e514d10d17
	round[3].diff_lay	e8896cfe45cec75347942124d535201f
4	round[4].start	e8896cfe45cec75347942124d535201f
	round[4].key_add	4959a3ea2f2aed5e1a39bf54d50efedf
	round[4].s_box	a41e0af74e56551243aa085cb562bb6c
	round[4].diff_lay	dc7e886d30d2c974e0ebb305103d062b
5	round[5].start	dc7e886d30d2c974e0ebb305103d062b
	round[5].key_add	049b11df7d7033f59dbb20a26644356e
	round[5].s_box	f23ee32dfffb766b35e8a54d233acd9ca
	round[5].diff_lay	15cad8053921e366b22527e263ac1655
6	round[6].start	15cad8053921e366b22527e263ac1655
	round[6].key_add	aa0a3c529739c399cc2bdc5fe377332b
	round[6].s_box	62dbeb8885aa2e02271786534d19c3ae
	round[6].diff_lay	c99046c53a2a4e5d0fe0777d157e4210
뒷 페이지에서 계속		

전 페이지 이어 계속		
라운드	함수	값
7	round[7].start	c99046c53a2a4e5d0fe0777d157e4210
	round[7].key_add	25d3bf894f4658f0bd4ca1360cc887c8
	round[7].s_box	3feff4f884ad5e257af9f164fe35ea4a
	round[7].diff_lay	7e4b14fdd83b2d9cff53fa40030299f3
8	round[8].start	7e4b14fdd83b2d9cff53fa40030299f3
	round[8].key_add	feb32a69c4fe63aacf87fa08d942b124
	round[8].s_box	0cf6e58e88acfb395f612d62e576c855
	round[8].diff_lay	7dfe7a68e5210f2d45e2409637446d10
9	round[9].start	7dfe7a68e5210f2d45e2409637446d10
	round[9].key_add	861aed854840260e47fc524654ec2251
	round[9].s_box	442e5336522d2362a0af480b20d694c2
	round[9].diff_lay	0af1fe0ac7fb646606dbdf4eb80d3d28
10	round[10].start	0af1fe0ac7fb646606dbdf4eb80d3d28
	round[10].key_add	ea0c01a4ea82833c549bccf94844ac80
	round[10].s_box	bb727ca9bbcddec0ffdde4bded4dd91ec
	round[10].diff_lay	91a588a05488541d077338fa399af522
11	round[11].start	91a588a05488541d077338fa399af522
	round[11].key_add	824cef10a95d7239be08d50cd05a5da1
	round[11].s_box	13f9613ed3a41eaaae62b57270b88de9
	round[11].diff_lay	0a3a6aef6ea7c7cd77df1ab94c3ac71d
12	round[12].start	0a3a6aef6ea7c7cd77df1ab94c3ac71d
	round[12].key_add	05cfa5f7b56a6d05113576150c7ee21b
	round[12].s_box	364506f8d20a3cc2e3c8389781a89803
	round[12].diff_lay	0d678661d5042bdcf0cb803f6343d143
13	round[13].start	0d678661d5042bdcf0cb803f6343d143
	round[13].key_add	60ba3d0d03eb052d64ad4356f76fc1b9
	round[13].s_box	d0a38b097bb036404376647568b1dd7d
	round[13].diff_lay	1d5bea5d7d63d073e0358b7a3539ccb9
14	round[14].start	1d5bea5d7d63d073e0358b7a3539ccb9
	round[14].key_add	e35ace4a123ab14d5f7a3007f4232886
	round[14].s_box	4de58bc83938c82f845104ccba893484
	round[14].diff_lay	5177e22f06f18f9eb71a3a8aae6b2d6b
15	round[15].start	5177e22f06f18f9eb71a3a8aae6b2d6b
	round[15].key_add	34a4be36219897617762d7a4b65732f8
뒷 페이지에서 계속		

전 페이지 이어 계속		
라운드	함수	값
16	round[15].s_box	18a95a64fd3b85e8f50e0d5d4e28a1f7
	round[15].diff_lay	6ecb38128765cf8676bdaaca83c1483a
	round[16].start	6ecb38128765cf8676bdaaca83c1483a
	round[16].key_add	19e7f69edc259a7b03aba1e52ccdd95f
	round[16].s_box	8e2e4292939db8d9d571325a424f3553
	round[16].output	00112233445566778899aabbccddeeff

참고 문헌

- [1] Kazumaro Aoki, Tetsuya Ichikawa, Masayuki Kanda, Mitsuru Matsui, Shiho Moriai, Junko Nakajima, and Toshio Tokita, *Camellia: A 128-bit block cipher suitable for multiple platforms - design and analysis*, LNCS 2012 , pages 39–56. Springer, 2000.
- [2] Joan Daemen and Vincent Rijmen, *The Design of Rijndael*. Springer, 2001.
- [3] J. Daemen, V. Rijmen, *AES proposal: Rijndael (2nd Version)*, AES Submission, 1999.
- [4] A. Menezes, P. van Oorshot, S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, New York, 1997, pp.81–83.
- [5] NIST FIPS PUB 46: Data Encryption Standard, January, 1977.
- [6] NIST FIPS PUB 197: Advanced Encryption Standard, November, 2001.
- [7] NESSIE Project, New European Schemes for Signatures, Integrity and Encryption, Homepage-available at <http://cryptonessie.org>.