ARIA

https://youtu.be/4GUkfCGaPgo

ARIA?

- 경량환경 & 하드웨어 구현을 위해 최적화된 Involutional SPN 구조를 갖는 범용 알고리즘 **(차세대 국가 암호화 알고리즘)**

- Involution 구조 : 암호화 과정과 복호화 과정이 같은 구조
- SPN(Substitution-Permuation-Networks) 구조 : S-box와 확산 함수가 반복적으로 사용되는 구조

ARIA 의 역사

- 2004년 국가 표준 암호 지정
- 2010년 국제 표준
- 현재, 국내에서는 주로 사용, 해외에서는 사용 x
- ARIA(Academy, Resarch Institute, Agency) 의 함축어의 약어

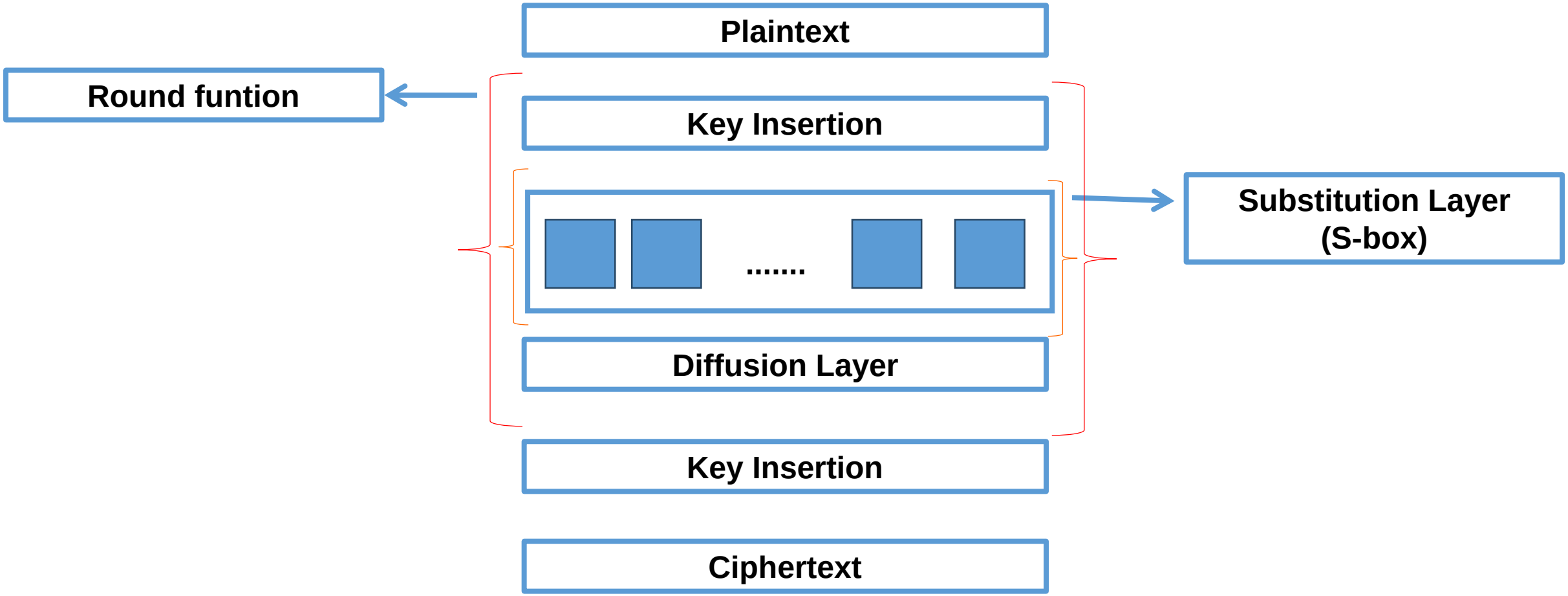
ARIA의 특징

- 블록 크기 : 128bit
- 키 크기 :128 / 192 / 256 bit **(AES 동일 규격)**
- 전체 구조 : Involutional Substitution-Permutation Network (Involutional SPN)
- 라운드 수 : 12 / 14 / 16 (키 크기에 따라 결정 됨)

라운드 수 키	128 bit	192 bit	256 bit
라운드	**12**	**14**	**16**

- 간단한 연산 사용 → 초경량 환경에 효율적
- 바이트 단위의 연산 → 하드웨어에 효율적

SPN


Plaintext
Round funtion
Key Insertion
.....
Substitution Layer
(S-box)
Diffusion Layer
Key Insertion
Ciphertext

ARIA의 특징

- 8 비트 환경과 하드웨어 구현에 뛰어난 효율성
- ics-card , VPN 장비 등 다양한 환경에서 적용 가능
- 소프트웨어 구현에서 Camellia보다 빠르고 AES에 근접하는 성능 보임

CPU	ARIA	AES	Camellia	SEED
Pentium 3	37.3	23.3	33.4	42.4
Pentium 4	49.0	30.5	83.9	81.3

SEED?

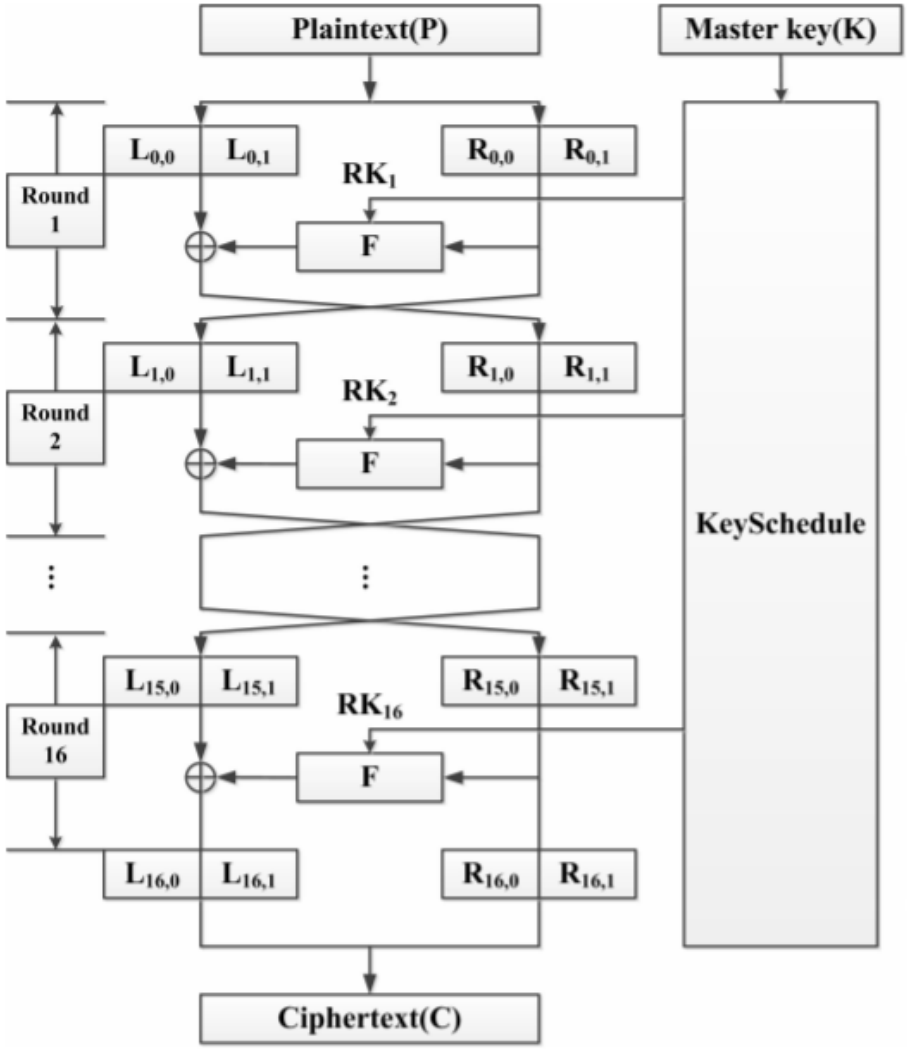
- 전자상거래, 금융, 무선 통신 등에서 전송되는 개인정보와 같은 중요 정보를 보호하기 위해 개발
- 1999.2 한국인터넷진흥원과 국내 암호 전문가들이 순수 국내 기술로 개발
- 1999년 SEED128 2009년 SEED 256 대칭키 블록 암호 알고리즘

SEED의 특징

- 블록 암호 알고리즘
- Feistel 구조
- 128bit의 평문 블록과 128bit 키를 입력으로 사용하여 총 16라운드를 거쳐 128bit 암호문 블록을 출력

SEED 구조도

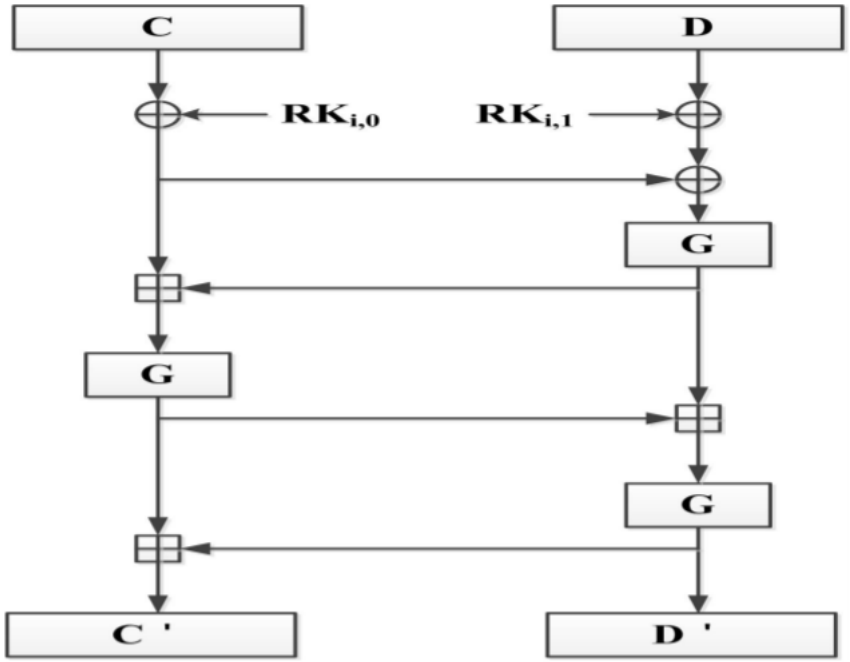
- 전체 구조도



SEED 구조도

- F 함수 (SEED's)

- $C' = G[G[G\{(C \oplus RK_{i,0}) \oplus (D \oplus RK_{i,1})\} \boxplus (C \oplus RK_{i,0})] \boxplus G\{(C \oplus RK_{i,0}) \oplus (D \oplus RK_{i,1})\}] \boxplus G[G\{(C \oplus RK_{i,0}) \oplus (D \oplus RK_{i,1})\} \boxplus (C \oplus RK_{i,0})]$
- $D' = G[G[G\{(C \oplus RK_{i,0}) \boxplus (D \oplus RK_{i,1})\} \boxplus (C \oplus RK_{i,0})] \boxplus G\{(C \oplus RK_{i,0}) \boxplus (D \oplus RK_{i,1})\}]$



RK_i : i 라운드 암·복호화키

$\oplus$: 배타적 논리합 연산 (XOR)

$\boxplus$: 법 2^{32}에서의 덧셈

https://www.researchgate.net/publication/263625946_SEED_and_ARIA_algorithm_design_methods_using_GEZEL

SEED 구조도

- G 함수 (SEED's)

$$m_0 = 0xfc, m_1 = 0xf3, m_2 = 0xcf, m_3 = 0x3f$$

$$Y_3 = S_2(a_3),\ Y_2 = S_1(a_2),\ Y_1 = S_2(a_1),\ Y_0 = S_1(a_0)$$

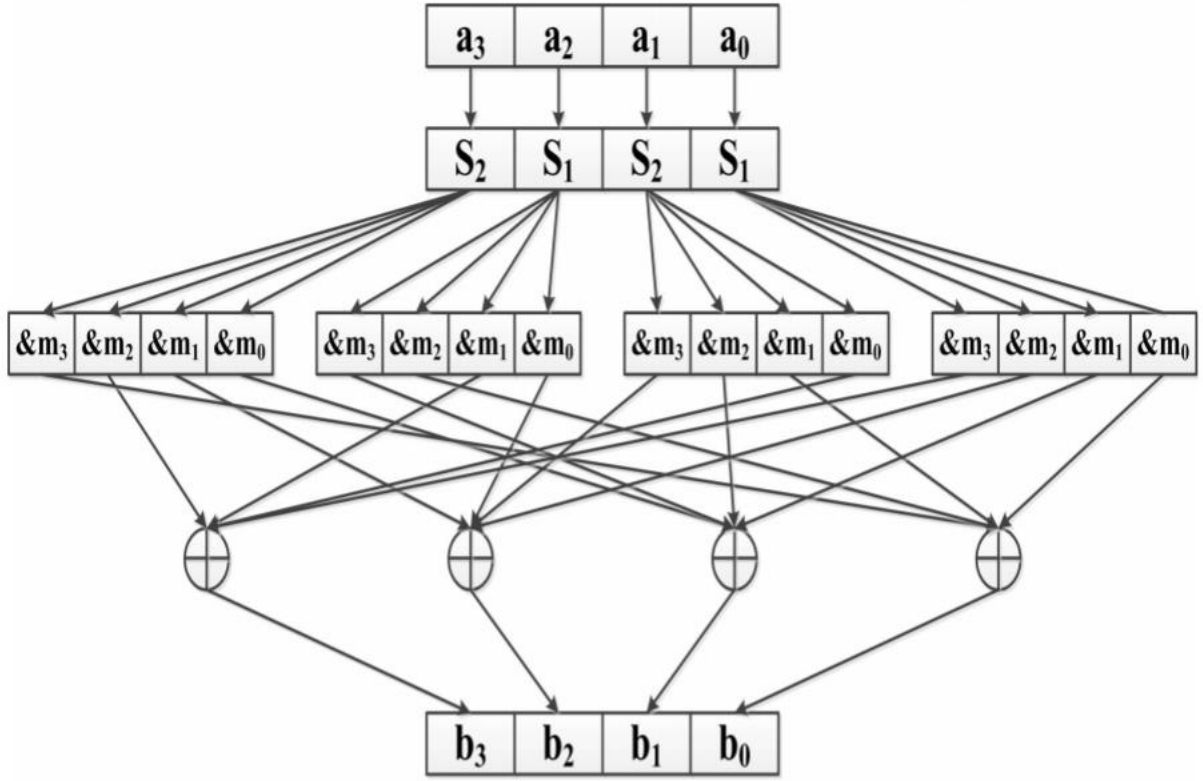
$$b_3 = (Y_0 \& m_3) \oplus (Y_1 \& m_0) \oplus (Y_2 \& m_1) \oplus (Y_3 \& m_2)$$

$$b_2 = (Y_0 \& m_1) \oplus (Y_1 \& m_2) \oplus (Y_2 \& m_3) \oplus (Y_3 \& m_0)$$

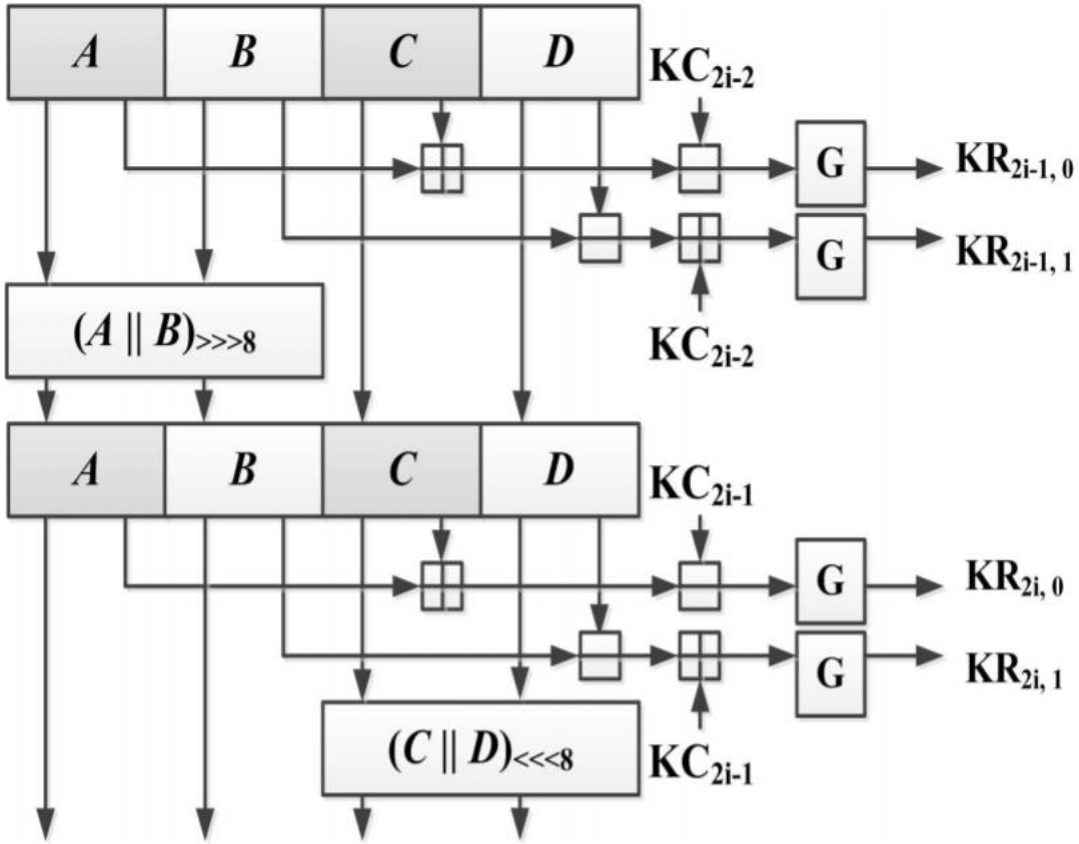
$$b_1 = (Y_0 \& m_2) \oplus (Y_1 \& m_3) \oplus (Y_2 \& m_0) \oplus (Y_3 \& m_1)$$

$$b_0 = (Y_0 \& m_0) \oplus (Y_1 \& m_1) \oplus (Y_2 \& m_2) \oplus (Y_3 \& m_3)$$

$\oplus$: 배타적 논리합 연산(XOR)



SEED 키 스케줄링



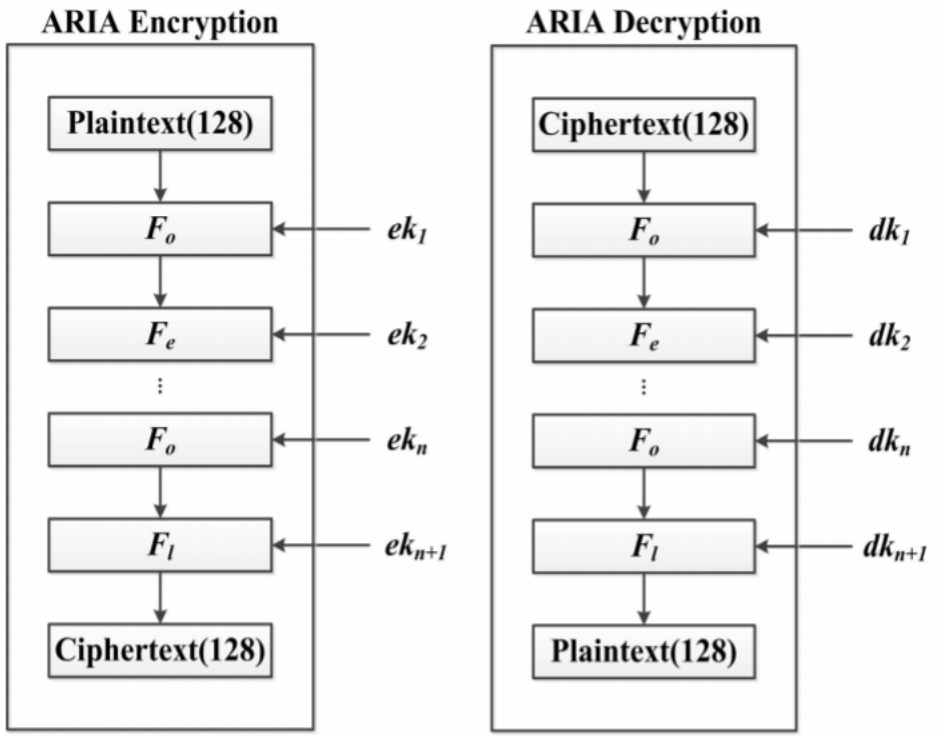
⊕ : 배타적 논리합 연산(XOR)
⊞ : 법 2^{32}에서의 덧셈
⊟ : 법 2^{32}에서의 뺄셈
RK_i : i라운드 암·복호화키

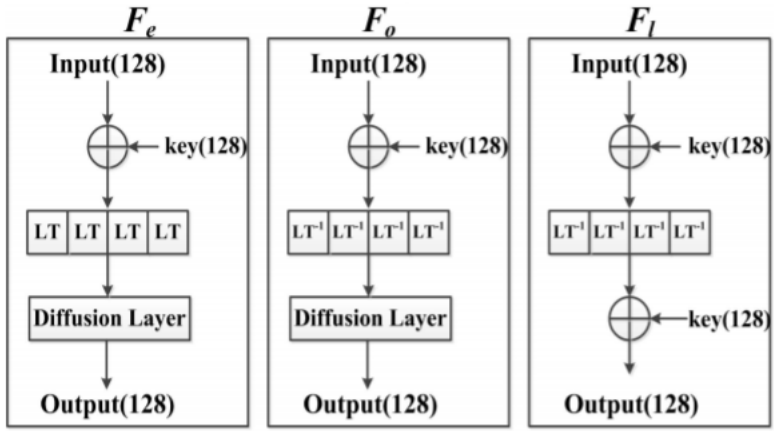
ARIA의 라운드 함수

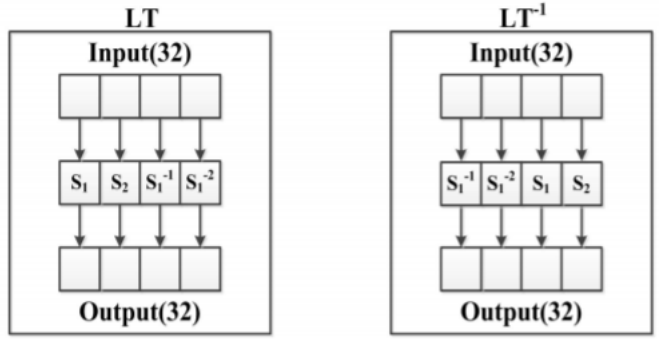
- 라운드 키 덧셈 : 128 비트 라운드 키를 라운드 입력 128 비트와 비트별 XOR 연산
- 치환 계층 : 두 유형의 치환 계층 존재, 각각 2종의 8비트 입출력 S-box와 그들의 역변환으로 구성
- 확산 계층 : 간단한 16 X 16 involution 이진 행렬을 사용한 바이트 간의 확산 함수로 구성

ARIA 구조

- 전체 구조


ARIA Encryption
Plaintext(128)
F_o
ek_1
F_e
ek_2
F_o
ek_n
F_l
ek_{n+1}
Ciphertext(128)
ARIA Decryption
Ciphertext(128)
F_o
dk_1
F_e
dk_2
F_o
dk_n
F_l
dk_{n+1}
Plaintext(128)


F_e
Input(128)
key(128)
LT LT LT LT
Diffusion Layer
Output(128)
F_o
Input(128)
key(128)
LT^{-1} LT^{-1} LT^{-1} LT^{-1}
Diffusion Layer
Output(128)
F_l
Input(128)
key(128)
LT^{-1} LT^{-1} LT^{-1} LT^{-1}
key(128)
Output(128)


LT
Input(32)
S_1 S_2 S_1^{-1} S_1^{-2}
Output(32)
LT^{-1}
Input(32)
S_1^{-1} S_1^{-2} S_1 S_2
Output(32)

ARIA 구조

- 치환 계층 (Substitution Layer)
  - 입력 값에 대응되는 S-box에 내부 상태 값을 출력해주는 비선형 대치 연산
  - S_1, S_2, S_1^{-1}, S_2^{-1} 4개의 S-box존재

ARIA 구조

- 확산 계층(Diffusion Layer)
  - 16개의 state값들을 행렬 곱 연산을 통하여 뒤섞는 변환
  - involution 성질을 만족하는 16x16 이진 행렬 A 사용
    - ex) 확산 함수의 입력값 (a0,.....a15)
      확산 함수의 출력값 (b1,.....b15)

✓ A가 고정 → 입력값과 관계없이
출력값을 구성하는 연산은 고정

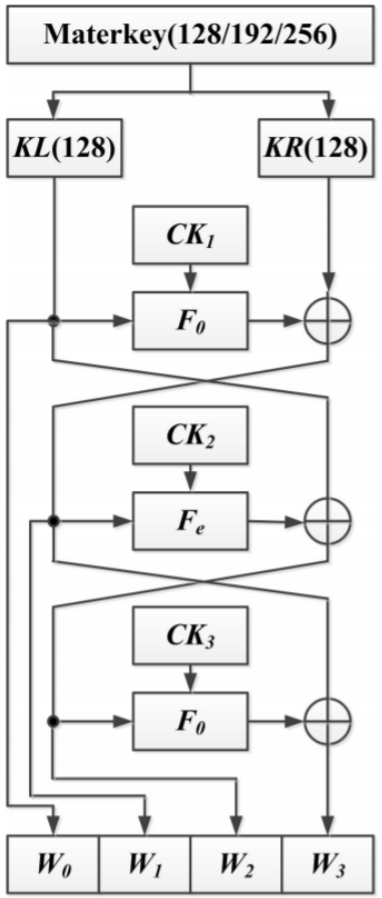
$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \\ b_9 \\ b_{10} \\ b_{11} \\ b_{12} \\ b_{13} \\ b_{14} \\ b_{15} \end{pmatrix} = \begin{pmatrix} 0&0&0&1&1&0&1&0&1&1&0&0&0&1&1&0 \\ 0&0&1&0&0&1&0&1&1&1&0&0&1&0&0&1 \\ 0&1&0&0&1&0&1&0&0&0&1&1&1&0&0&1 \\ 1&0&0&0&0&1&0&1&0&0&1&1&0&1&1&0 \\ 1&0&1&0&0&1&0&0&0&0&0&1&0&0&1&1 \\ 0&1&0&1&1&0&0&0&0&0&1&0&0&0&1&1 \\ 1&0&1&0&0&0&0&1&0&1&0&0&1&1&0&0 \\ 0&1&0&1&0&0&1&0&1&0&0&1&1&1&0&0 \\ 1&1&0&0&1&0&0&1&0&0&1&0&0&1&0&1 \\ 1&1&0&0&0&1&1&0&0&0&0&1&1&0&1&0 \\ 0&0&1&1&0&1&1&0&1&0&0&0&0&1&0&1 \\ 0&0&1&1&0&0&0&1&0&0&0&0&1&0&1&0 \\ 0&1&1&0&0&0&1&1&0&0&0&1&1&0&0&0 \\ 1&0&0&1&0&0&1&1&1&0&1&0&0&1&0&0 \\ 1&0&0&1&1&1&0&0&0&1&0&1&0&0&1&0 \\ 0&1&1&0&1&1&0&0&1&0&1&0&0&0&0&1 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{15} \end{pmatrix}$$

ARIA 구조

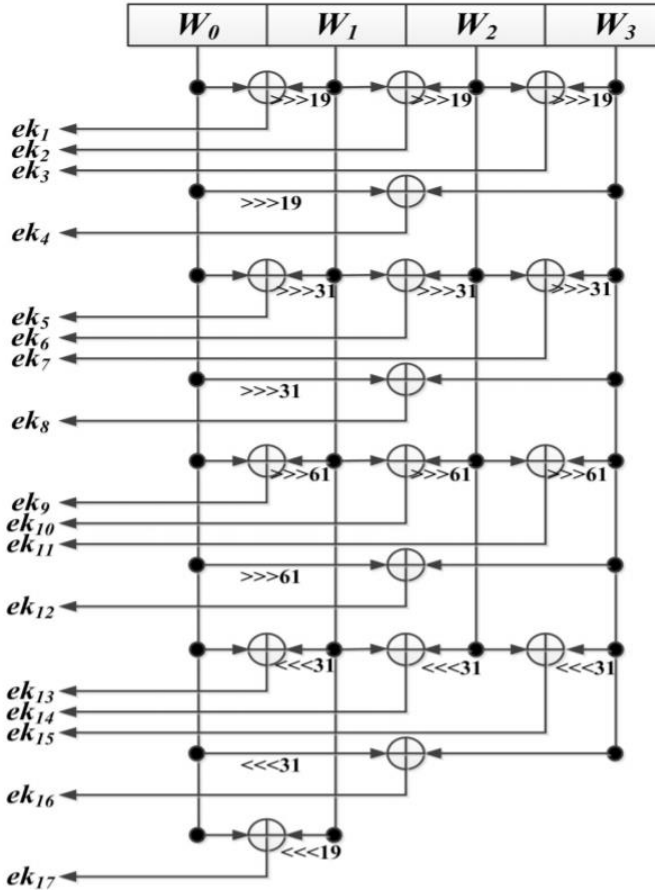
- 키 스케줄링

$$W_0 = KL$$
$$W_1 = F_o(W_0 \oplus CK_1) \oplus KR$$
$$W_2 = F_e(W_1 \oplus CK_2) \oplus KL$$
$$W_3 = F_o(W_2 \oplus CK_3) \oplus W_1$$

F_o : 홀수 라운드 함수
F_e : 짝수 라운드 함수
F_l : 마지막 라운드 함수
ek_i : i라운드 암호화키
dk_i : i라운드 복호화키



키 초기화 과정



ARIA 키 스케줄링

https://www.researchgate.net/publication/263625946_SEED_and_ARIA_algorithm_design_methods_using_GEZEL

Q & A