

Block cipher

SeoulTech Cryptography & Information Security Lab.

조민정

2019-07-12

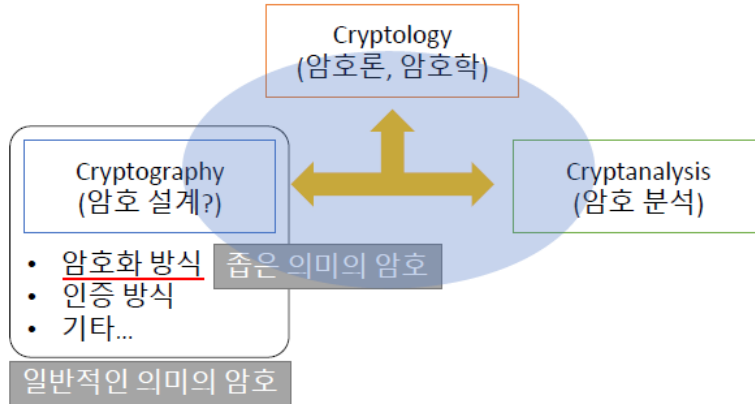
Contents

- 1 Review
- 2 블록 암호 개요
- 3 SPN 구조
- 4 Feistel 구조
- 5 ARIA 알고리즘
- 6 과제

지난 시간에 무엇을 배웠나요?

Cryptography란?

-



Cryptography & Information Security

- Cryptography는 Information Security의 핵심 도구

Cryptosystem

- 비밀키 암호 : 키가 공개되어 있지 않음, 블록암호, 스트림암호 등
- 공개키 암호 : 키가 공개되어 있음, 인수분해 기반 암호, 양자 암호 등
- 하이브리드 시스템 : 공개키 암호 시스템 + 대칭키 암호 시스템

블록 암호 개요

Kerckhoff의 원리



Auguste Kerckhoff는 누구인가?

- 19세기의 네덜란드 암호학자
- 'Le Journal des Sciences Militaires' 저서로 유명
- 군사적 암호에 대한 원칙을 기술
- 6가지 군사 암호를 위한 암호 설계 원칙을 제시
 1. 시스템은 수학적이거나 해독할 수 없지 않고 실제적이여야한다.
 2. 내용을 비밀로 하지 않아야 하며, 내용이 적에게 알려져도 상관이 없어야한다.
 3. 서면이 아닌 방법으로 통신이 가능하고 보관 가능해야 하며 사용자가 수정 가능해야한다.
 4. 전신 통신에 적용 가능해야 한다.
 5. 장치와 문서는 휴대 가능해야하며 사용법과 기능에 여러 사람이 필요하지 않아야 한다.
 6. 적용되는 환경에서 시스템은 사용하기 쉬어야 하며 정신적 긴장감이나 규칙을 이해하기 위한 지식이 요구되지 않아야 한다.

블록 암호 개요

DES의 논쟁

DES란?

1977년 미국 표준 기술 연구소 (NIST)의 전신인 미국 표준국(NBS)에 의해 미 연방 정보처리 표준 46(FIPS PUB46)으로 채택된 블록 암호 알고리즘

논쟁1. 키길이가 56bit로 전수조사에 취약하다.

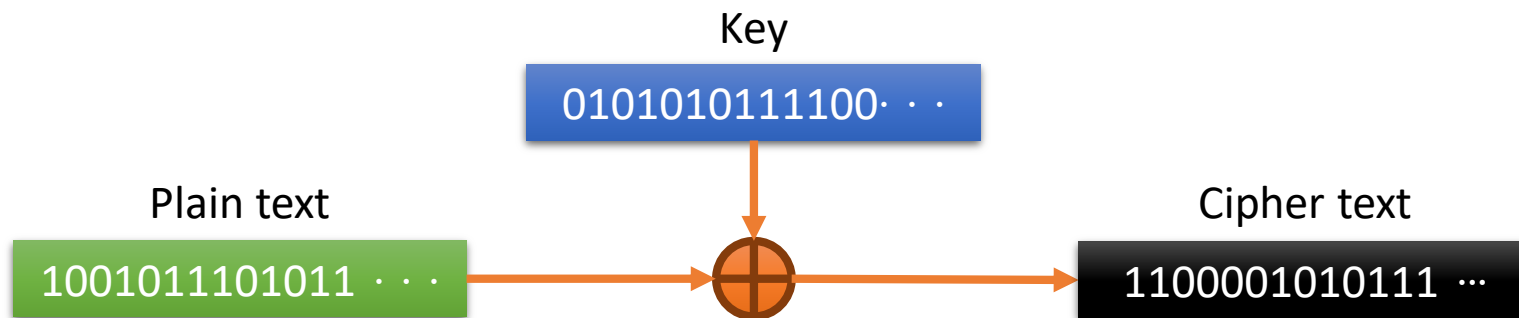
논쟁2. DES 내부 구조 설계 기준인 S-box가 비밀로 지정됐으며 아직도 비밀 사항이다.

블록 암호 개요

One-time pad

One-time pad

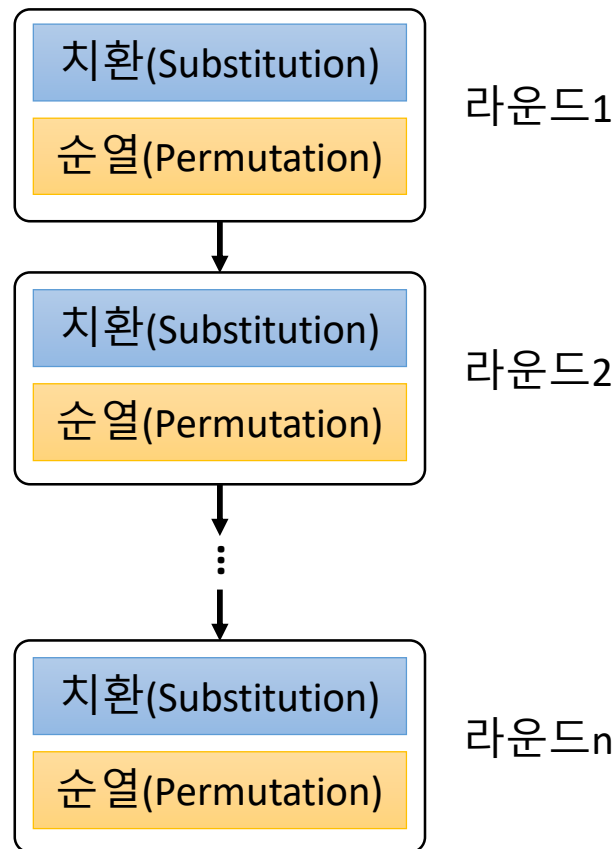
- 완벽한 보안이라 여겨짐
- 평문 각 비트와 독립적인 키 비트들을 xor 하는 방식
- 암호화할 평문 비트 만큼 키 비트가 필요함



- 그러나 암호화하는 비트가 늘어나며 같은 패턴의 키를 쓰게 되고 이는 보안성에 문제를 가짐
- Ex) Vernam cipher

블록 암호 개요

Block cipher structure – 암호 구조



블록 암호?

- 평문 블록 전체를 가지고 같은 크기의 암호문 블록 생성
- 암호화가 역이 성립해야함
- 평문 블록에 대하여 유일한 암호문 블록을 생성해야함.
- 혼돈과 확산의 개념을 통해 통계적 분석에 기초한 암호 해독을 방지하고자 노력함.

혼돈

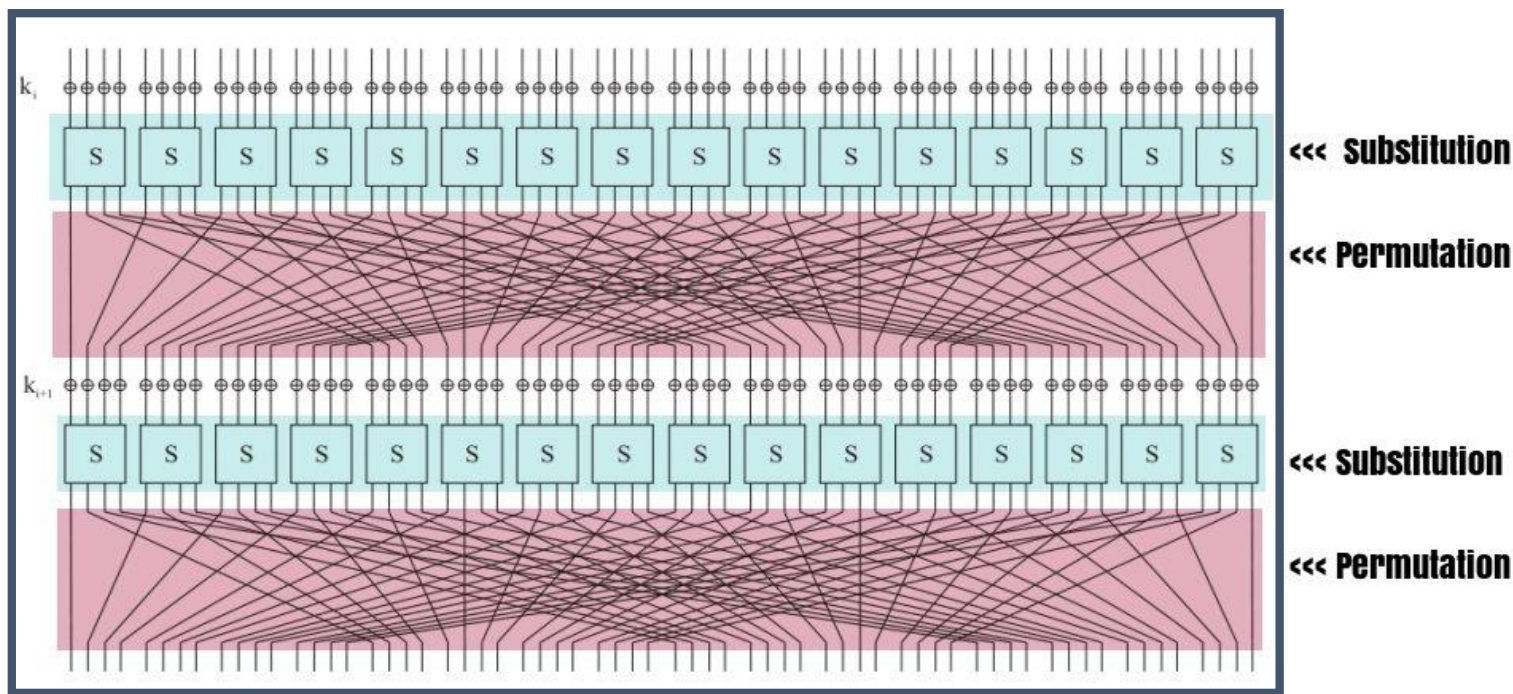
- 암호문에 대한 통계값과 암호 키 값 사이의 관계를 가능한 복잡하게 함
- 키를 발견하기 어렵게 하기 위함
- 복잡한 치환이용
- 단순 선형 치환 함수는 혼돈의 효과가 미비함

확산

- 평문의 통계적 구조가 암호문의 광범위한 통계 값에 분산되는 방식
- 암호문 비트들은 여러 평문 비트에 영향을 받음

블록 암호 개요

Block cipher structure – 치환과 순열



치환 (Substitution)

평문의 각 원소 또는 원소의 그룹을 다른 원소에 사상 시킴

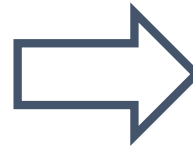
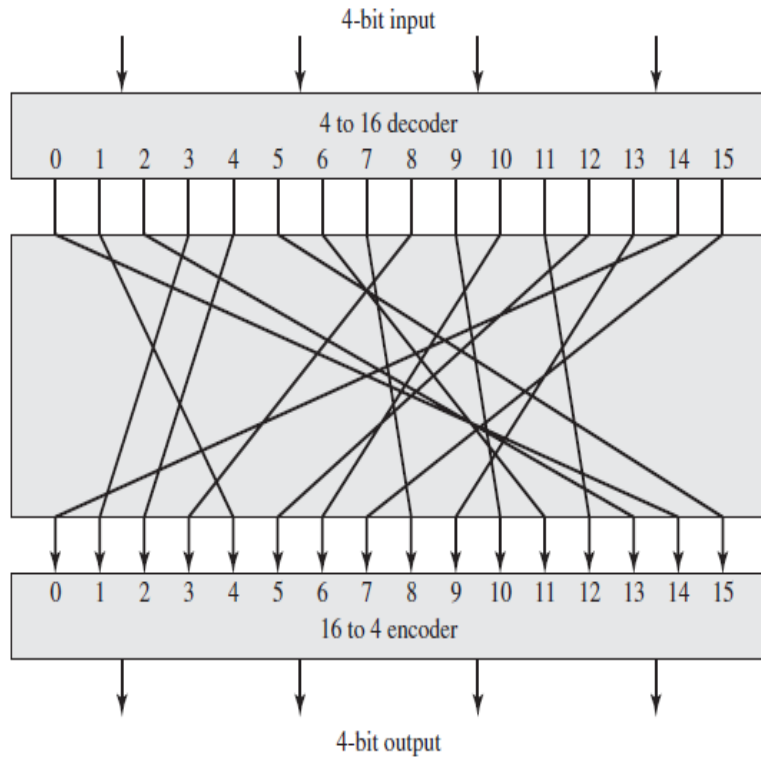
순열 (Permutation)

평문 원소의 순서는 순열의 순서대로 재배치됨

원소는 추가 및 삭제 또는 재배치되며 순열의 순서대로 변경됨

블록 암호 개요

Block cipher structure – 치환과 순열

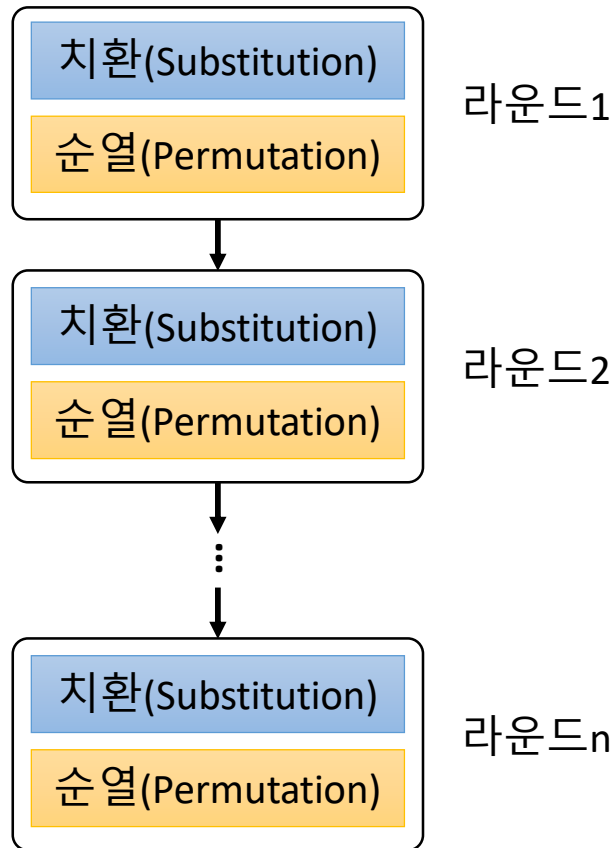


Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

Ciphertext	Plaintext
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

블록 암호 개요

Block cipher structure – 암호 구조



선형

- 대수적 방정식이 선형방정식의 형식을 갖추
- 기하학적 비례,모양, 형태가 직선적임
- 중첩의 원리(비례성, 가산성)를 따름
- Permutation

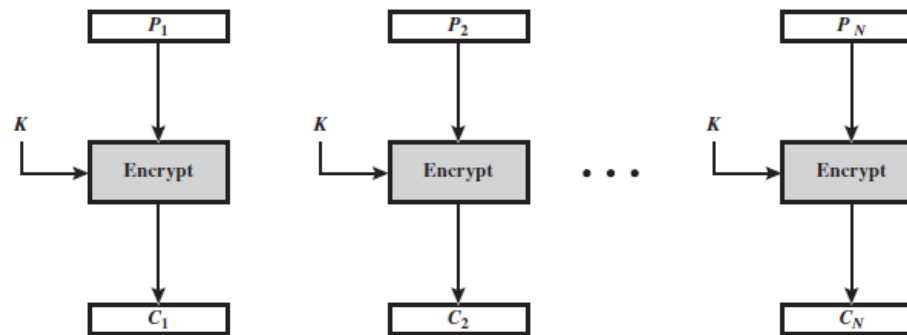
비선형

- 중첩의 원리를 만족하지 않음
- Substitution
- Modular addition

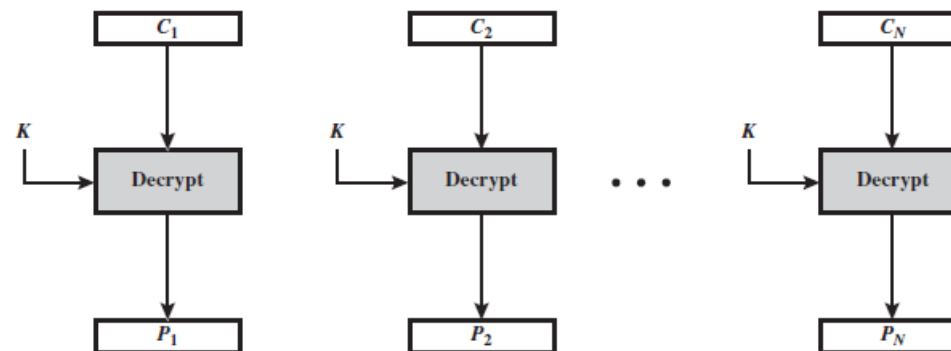
블록 암호 개요

Block cipher structure – 운용모드

평문이 엄청 긴 경우?



(a) Encryption

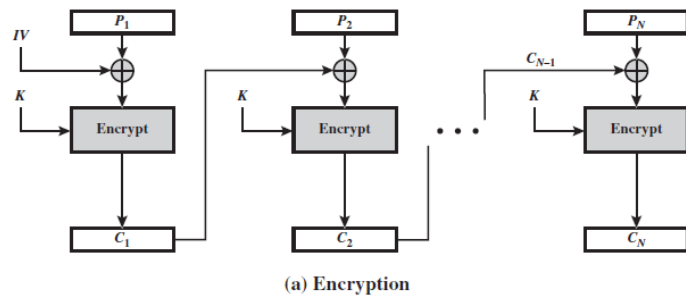


(b) Decryption

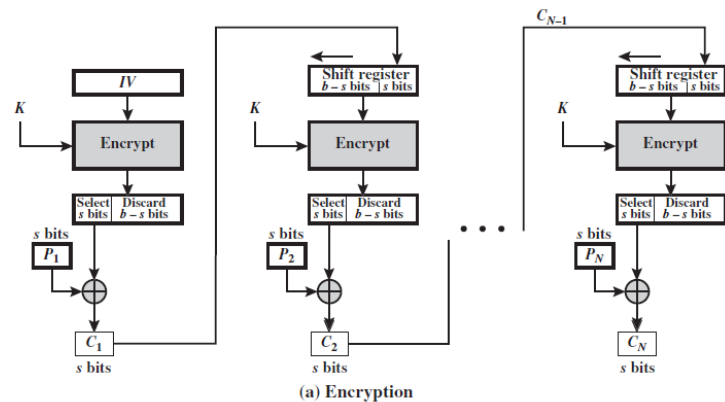
ECB(Electronic Codebook) 모드

블록 암호 개요

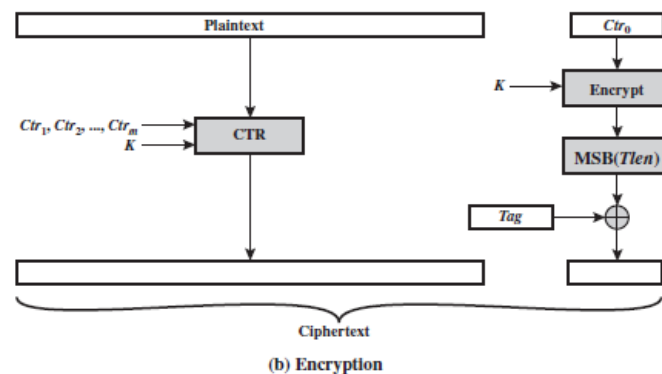
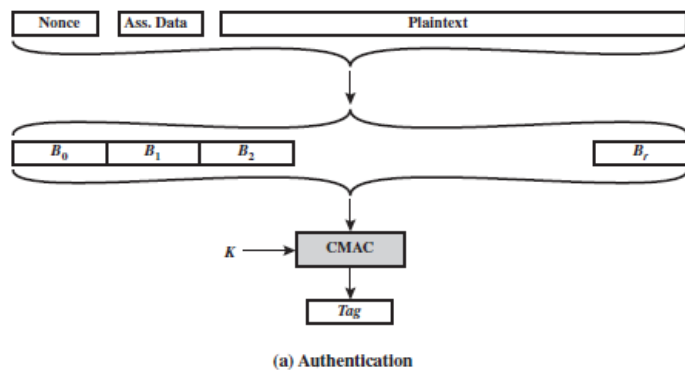
Block cipher structure – 운용모드



CBC(Cipher Block Chaining) 모드



CFB(Cipher FeedBack) 모드



CCM(counter with cipher block chaining Message Authentication Code) 모드

블록 암호 개요

Block cipher structure – 운용모드

평문이 한 블록 보다 짧은 경우?

Block 1	...	Block q
XX..XXX	...	XX..XX10...00

OZ-PAD

Block 1	...	Block q
XX..XXX	...	XX..XXR0...00

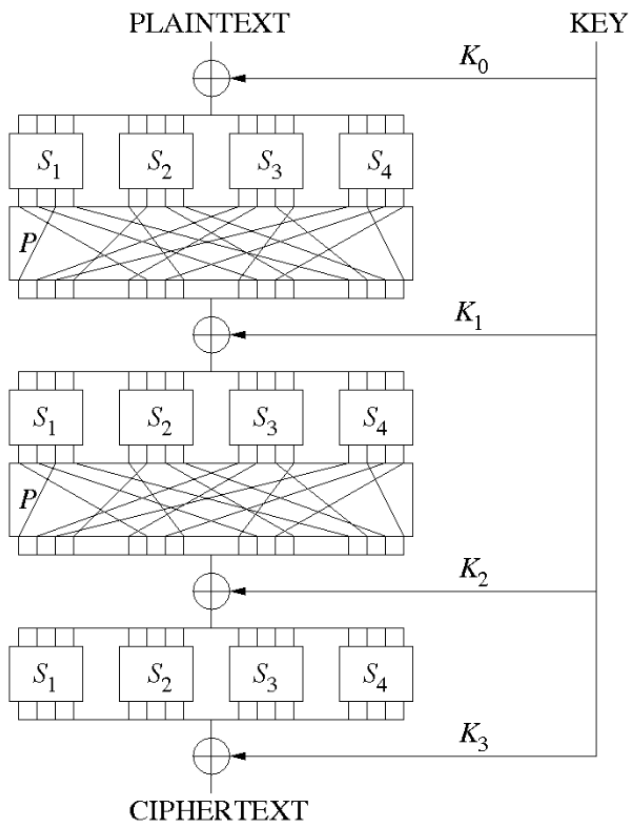
NIST-PAD

L_M Block	Block 1	...	Block q
$128*(q-1)+122$	XX..XXX	...	Xxxxxx .. X000000

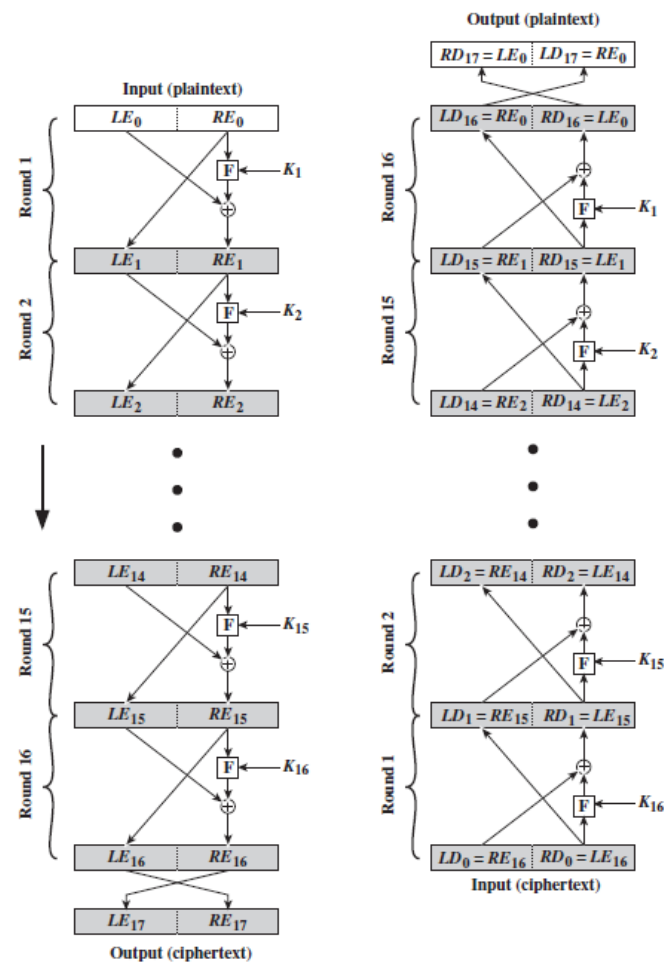
ISO(9797-1)

블록암호 구조

Block cipher structure – SPN structure , Feistel structure



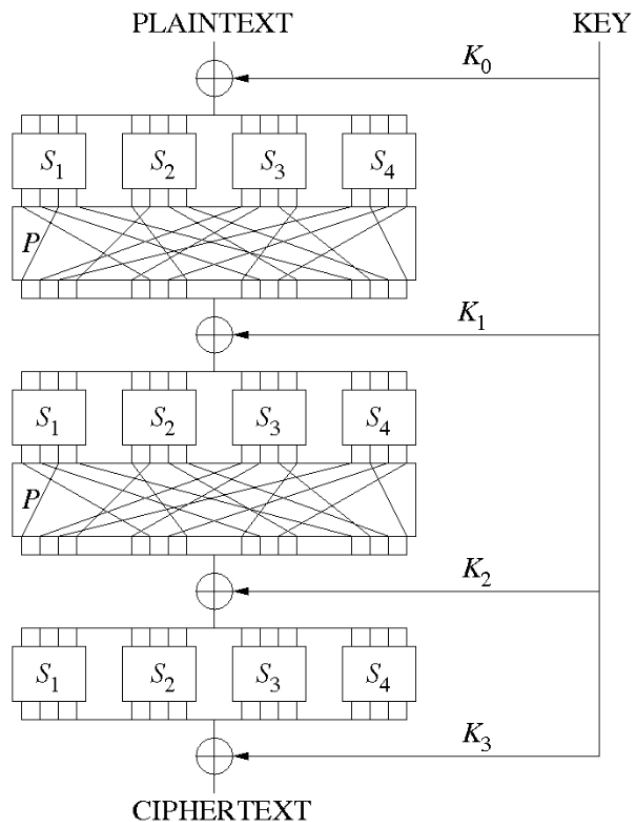
SPN Structure



Feistel Structure

SPN 구조

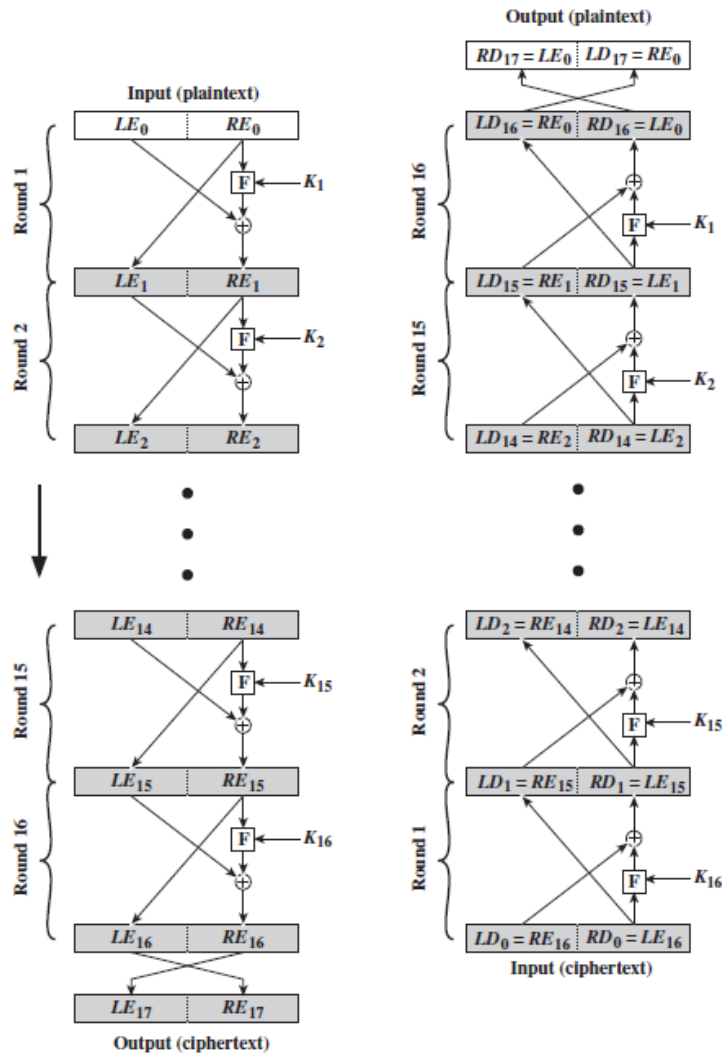
Block cipher structure – SPN structure



- SPN 구조의 구성
 - 혼돈을 수행하는 치환층
 - 확산을 수행하는 확산층
 - 키 xor
- 암호 알고리즘과 복호 알고리즘이 다름
- 각 라운드에서 모든 비트에 비선형변환이 일어남
 - 라운드 수가 Feistel 구조에 비해 적음
- SPN 구조 블록 암호
 - AES, ARIA

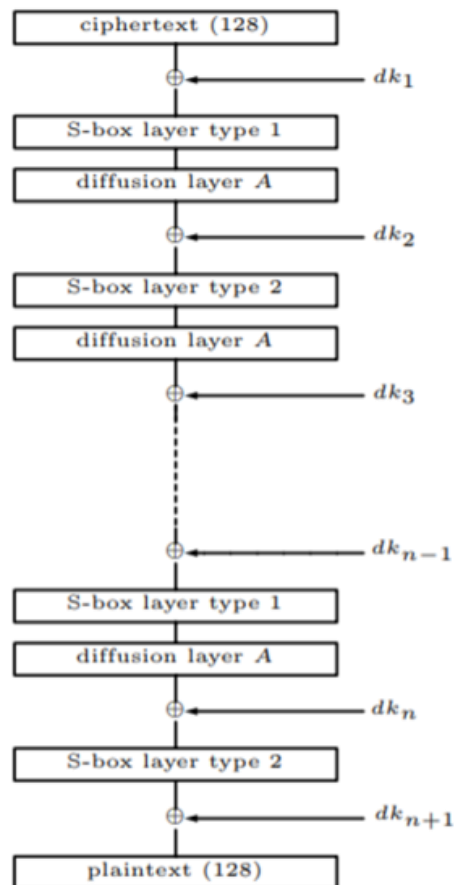
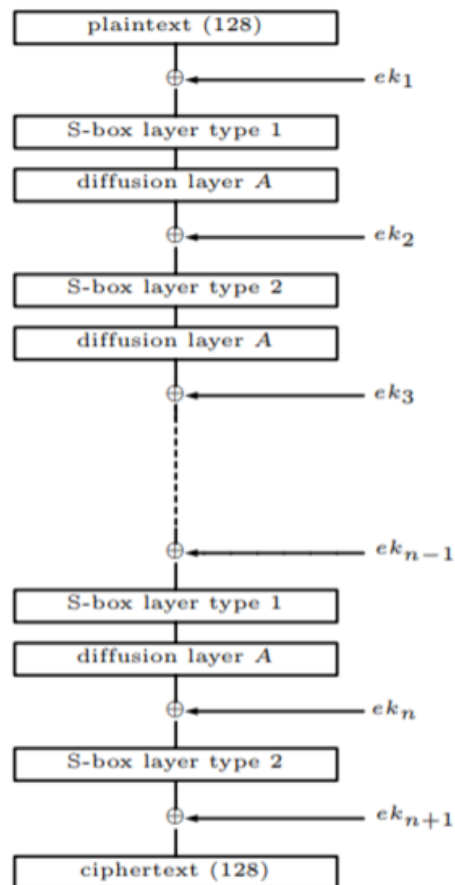
Feistel 구조

Block cipher structure – Feistel structure



- Feistel
 - SPN의 특별한 형태
- 순열(permutation), 치환(substitution)?
- 암호 알고리즘과 복호 알고리즘이 같음
- Feistel 구조 블록 암호
 - DES

ARIA 알고리즘



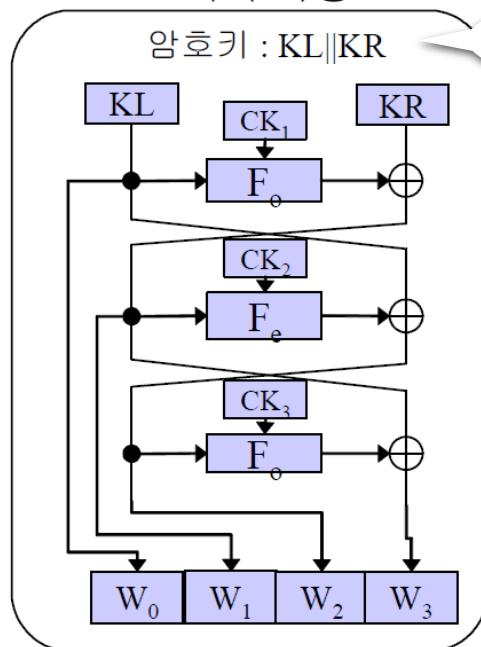
- 구조?
- 경량 환경 및 하드웨어 구현을 위해 최적화된 구조
- 블록크기 : 128 bit
- 키 크기 : 128/192/256 bit
- 라운드 수 : 12/14/16 라운드
: 키 크기에 종속적
- <https://seed.kisa.or.kr/kisa/Board/19/detailView.do>
- <https://tools.ietf.org/html/rfc5794>

ARIA 알고리즘

ARIA - 키 확장 & 라운드 키 생성

키 확장
초기화 과정

암호키 : $KL || KR$



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

$C1 = 0x517cc1b727220a94fe13abe8fa9a6ee0$

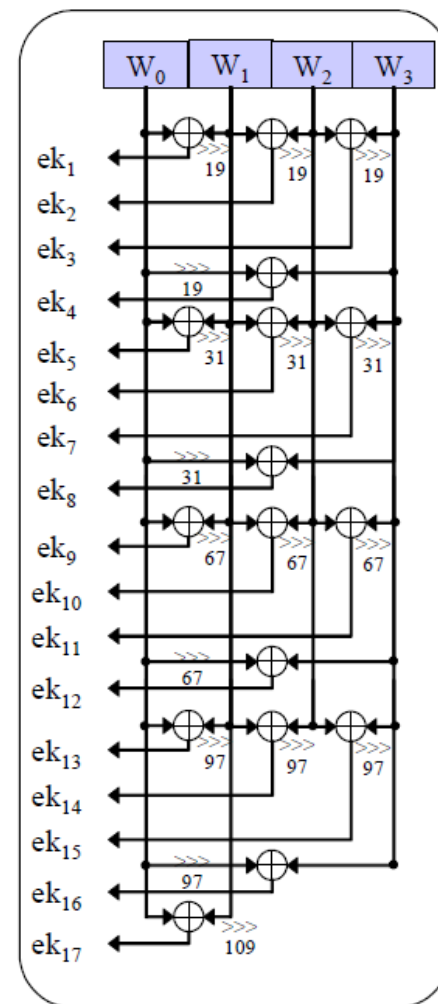
$C2 = 0x6db14acc9e21c820ff28b1d5ef5de2b0$

$C3 = 0xdb92371d2126e9700324977504e8c90e$

<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	$C3$	$C1$	$C2$

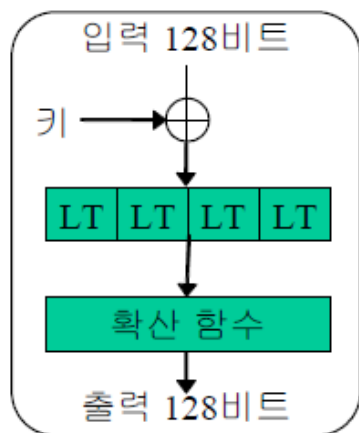
라운드 키 생성



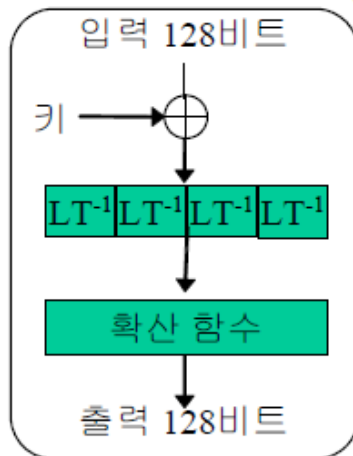
ARIA 알고리즘

ARIA - 라운드 함수

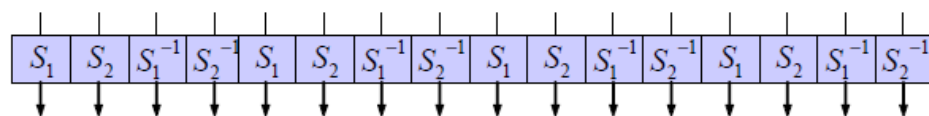
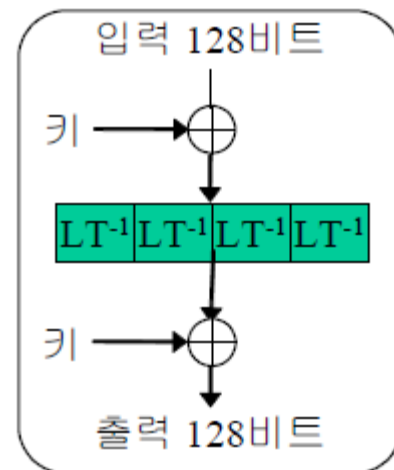
홀수 라운드 함수 F_o



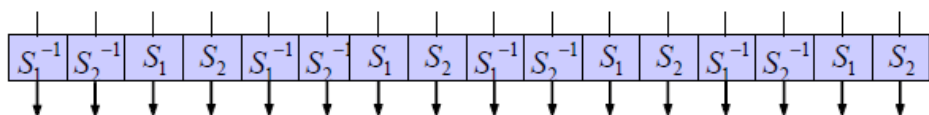
짝수 라운드 함수 F_e



최종 라운드 함수 F_f



치환 계층 (유형 1)



ARIA 알고리즘

ARIA – 치환(Substitution)

- Sbox : 8 bit input - 8bit output
- Ex) AB -> A행 B열

<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	ba	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

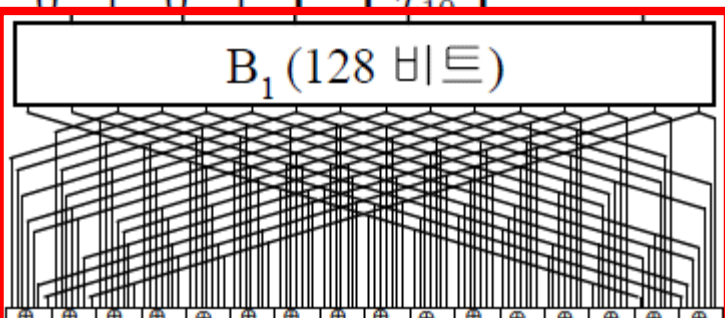
<표 7> S-box S_2^{-1}

	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

ARIA 알고리즘

ARIA – 순열(Permutation)

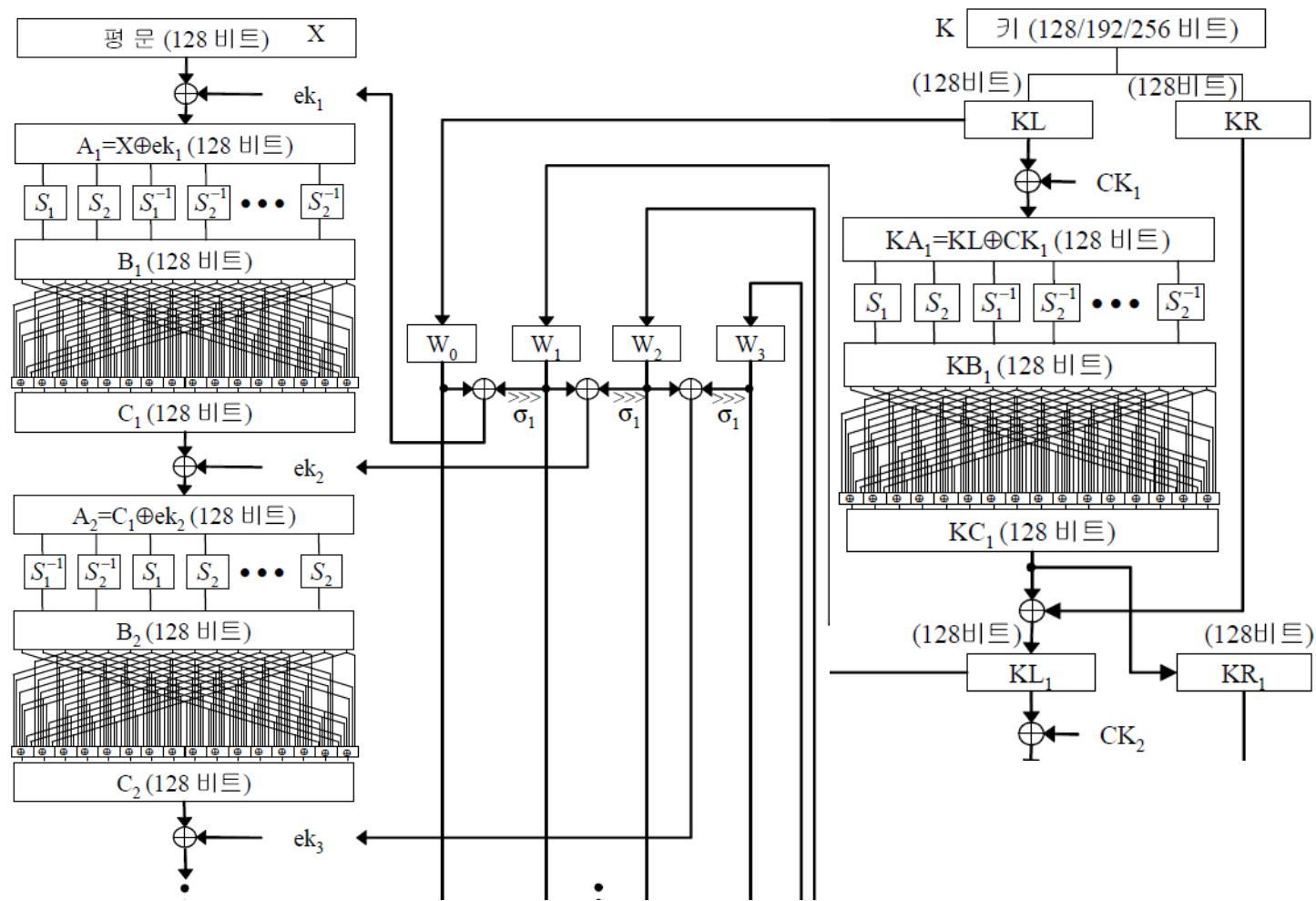
$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 &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



B₁ (128 비트)

C₁ (128 비트)

ARIA 알고리즘



ARIA 알고리즘

ARIA 고속구현

- ARIA 고속구현?
 - Substitution 과 Permutation을 한번에 할 수 있는 Lookup table 구성

- ARIA 치환의 설계?

- $M_1^{-1} \cdot M_2 \cdot M_1$, $M_1^{-1} = M_1$, $A = M_1 \cdot M_2 \cdot M_1 = M_1 \cdot P \cdot M \cdot M_1 = M_1 \cdot P \cdot M \cdot M_1 \cdot M \cdot M$
 $= M_1 \cdot P \cdot M \cdot M \cdot M_1 \cdot M = M_1 \cdot P \cdot M_1 \cdot M.$

$$M_1 = \begin{pmatrix} I & I & I & 0 \\ I & 0 & I & I \\ I & I & 0 & I \\ 0 & I & I & I \end{pmatrix}, \quad M_2 = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & P_1 & 0 & 0 \\ 0 & 0 & P_2 & 0 \\ 0 & 0 & 0 & P_3 \end{pmatrix} \cdot \begin{pmatrix} T & 0 & 0 & 0 \\ 0 & T & 0 & 0 \\ 0 & 0 & T & 0 \\ 0 & 0 & 0 & T \end{pmatrix},$$

$$T = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \quad P_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad P_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad P_3 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

$$P = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & P_1 & 0 & 0 \\ 0 & 0 & P_2 & 0 \\ 0 & 0 & 0 & P_3 \end{pmatrix}, \quad M = \begin{pmatrix} T & 0 & 0 & 0 \\ 0 & T & 0 & 0 \\ 0 & 0 & T & 0 \\ 0 & 0 & 0 & T \end{pmatrix}.$$

ARIA 구현

- 사용 언어 : c언어
- 제출 : 코드와 측정한 속도
- 제출 방법 : 홈페이지(추후 공지 예정)
- 제출 형태 : github 링크 및 설명
- 기한 : 2019년 7월 17일 수요일

- [1] http://www.ktword.co.kr/abbr_view.php?m_temp1=2632
- [2] http://www.ktword.co.kr/abbr_view.php?nav=&m_temp1=5417&id=142
- [3] Kwon, Daesung, et al. "New block cipher: ARIA." *International Conference on Information Security and Cryptology*. Springer, Berlin, Heidelberg, 2003.
- [4] https://en.wikipedia.org/wiki/Substitution%E2%80%93permutation_network#/media/File:SubstitutionPermutationNetwork2.png
- [5] 조경연. (2011). 암호와 복호가 동일한 SPN 블록 암호 SSB. 한국정보통신학회논문지, 15(4), 860-868.
- [6] 이선근, 정우열. (2005). 대용량 고속화 수행을 위한 변형된 Feistel 구조 설계에 관한 연구. 한국컴퓨터정보학회논문지, 10(3), 183-188.
- [7] 김기문, 박명서, 김종성, 이창훈, 문덕재, 홍석희. (2015). 패딩 오라클 공격에 따른 다양한 패딩방법의 안전성 분석. 정보보호학회논문지, 25(2), 271-278.
- [8] <https://tools.ietf.org/html/rfc5794>
- [9] <https://seed.kisa.or.kr/kisa/Board/19/detailView.do>
- [10] http://www.infosecwriters.com/Papers/Block_Cipher_Algorithms.pdf
- [11] Van Tilborg, Henk CA, and Sushil Jajodia, eds. *Encyclopedia of cryptography and security*. Springer Science & Business Media, 2014.
- [12] Petitcolas, Fabien. "La cryptographie militaire." (1883).
- [13] Stallings, William. *Network and internetwork security: principles and practice*. Vol. 1. Englewood Cliffs, NJ: Prentice Hall, 1995.

Thank you