Network Security Essentials

- 2장 대칭 암호와 메시지 기밀성(1) -

전 상 기(sanggi@pel.smuc.ac.kr)

상명대학교 프로토콜공학연구실

목 차

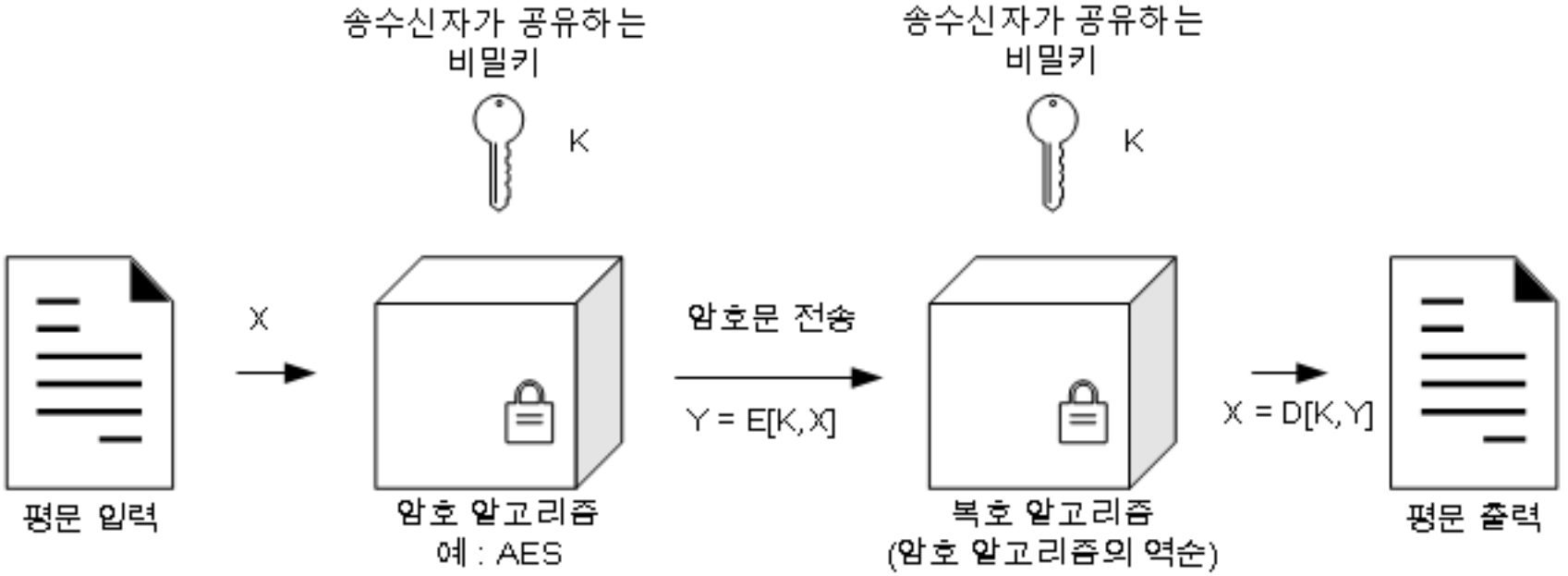
- 대칭 암호 원리
- 대칭 암호 알고리즘
- 랜덤넘버와 의사랜덤넘버

대칭 암호 원리

- 대칭 암호(Symmetric encryption) 구조
  - 평문(Plaintext) : 원문이나 데이터로서 알고리즘의 입력으로 이용
  - 암호 알고리즘(Encryption algorithm) : 원문을 다양한 방법으로 대체(Substitution)하고 치환(Transposition)하는 것
  - 비밀 키(Secret key) : 알고리즘의 한 입력으로 이용하며 정확한 대체와 치환이 이루어짐
  - 암호문(Ciphertext) : 출력으로 나오는 암호화된 메시지
  - 복호 알고리즘(Decryption algorithm) : 암호 알고리즘을 역으로 수행하는 것

대칭 암호 원리

- 안전하게 사용하는데 지켜야 할 두 가지 필수 사항
  - 강한 암호 알고리즘이 있어야 함
  - 송신자와 수신자는 공유하는 비밀 키를 안전한 방법으로 획득해야 하고 안전하게 보관해야 함
- 대칭 암호 단순 모델



대칭 암호 원리

- 암호
  - 암호해독(Cryptanalysis) : 평문이나 키를 찾으려는 시도
  - 전수 공격(Brute-force attack) : 모든 가능한 경우를 다 시도해보는 것
  - 암호 시스템의 일반적인 세 개의 독립적인 단계
    - 평문을 암호문으로 전환하는데 사용되는 연산 유형
      - 대체(Substitution)
      - 치환(Transposition)
    - 사용되는 키의 수
      - 동일한 키를 사용 : 대칭 암호(Symmetric encryption), 관용 암호(Conventional encryption), 비밀키 암호(Secret-key encryption), 단일키 암호(Single-key encryption)
      - 서로 다른 키를 사용 : 비대칭 암호(Asymmetric encryption), 쌍키 암호(Pairwise key encryption), 공개키 암호(Public key encryption)

대칭 암호 원리

- 암호
  - 암호 시스템의 일반적인 세 개의 독립적인 단계
    - 평문이 처리되는 방법
      - 블록 암호(block cipher) : 한 번에 한 블록 씩 입력하여 처리하고 한 블록씩 출력
      - 스트림 암호(stream cipher) : 연속적으로 처리하고 한번에 한 요소 씩 출력

대칭 암호 원리

- 암호화된 메시지 공격 유형
  - 암호문만 알고 있는 공격(Ciphertext-only attack)
    - 통계적 성질과 문자의 특성 등을 추정하여 해독
  - 알려진 평문 공격(Known-plaintext attack)
    - 공개된 평문/암호문 쌍을 이용하여 다음 암호문을 해독
  - 선택 평문 공격(Chosen-plaintext attack)
    - 알려진 평문 공격과 유사
      - 차이점 : 선택 평문 공격은 공격자에게 주어진 평문/암호문 쌍은 공격자가 선택한 값
  - 선택 암호문 공격(Chosen-ciphertext attack)
    - 암호문을 선택하고 그에 대응하는 평문을 얻는다는 점을 제외하고 선택 평문 공격과 유사함

대칭 암호 원리

- 암호화된 메시지 공격 유형
  - 암호화된 메시지 공격 유형 표

공격 유형	암호해독가가 알고 있는 정보
암호문만 알고 있는 공격(ciphertext-only attack)	• 암호 알고리즘 • 해독해야 할 암호문
알려진 평문 공격(known-plaintext attack)	• 암호 알고리즘 • 해독해야 할 암호문 • 비밀키로 만들어진 한 쌍 혹은 여러 쌍의 평문-암호문
선택 평문 공격(chosen-plaintext attack)	• 암호 알고리즘 • 해독해야 할 암호문 • 해독가가 선택한 평문 메시지와 비밀키로 그 평문을 암호화한 암호문
선택 암호문 공격(chosen-ciphertext attack)	• 암호 알고리즘 • 해독해야 할 암호문 • 해독가가 목적을 갖고 선택한 암호문과 비밀키로 그 암호문을 복호화한 평문
선택문 공격(chosen-text attack)	• 암호 알고리즘 • 해독해야 할 암호문 • 해독가가 선택한 평문 메시지와 키를 가지고 그 평문을 암호화한 암호문 • 해독가가 목적을 갖고 선택한 암호문과 비밀키로 그 암호문을 복호화한 평문

대칭 암호 원리

- 암호 구조가 계산적으로 안전한 구조
  - 암호문을 깨는데 드는 비용이 암호화된 정보의 가치보다 큼
  - 암호문을 깨는데 걸리는 시간이 해당 정보의 수명보다 김
- 키 탐색에 요구되는 평균 시간

키 크기(비트)	키의 종류 수	μs당 한 번의 암호화를 할 때 소요되는 시간	μs당 10^6번의 암호화를 할 때 소요되는 시간
32	$2^{32} = 4.3 \times 10^9$	2^{31}μs = 35.8분	2.15밀리초
56	$2^{56} = 7.2 \times 10^{16}$	2^{55}μs = 1142년	10.01시간
128	$2^{128} = 3.4 \times 10^{38}$	2^{127}μs = 5.4×10^{24}년	5.4×10^{18}년
168	$2^{168} = 3.7 \times 10^{50}$	2^{167}μs = 5.9×10^{36}년	5.9×10^{30}년
26개 문자(치환)	$26! = 4 \times 10^{26}$	2×10^{26}μs = 6.4×10^{12}년	6.4×10^6년

대칭 암호 원리

- Feistel 암호
  - 1973년 IBM의 Horst Feistel이 최초로 소개한 구조를 따라 만들어짐
  - 대부분의 대칭 블록 암호 알고리즘의 구조는 Feistel 암호 구조를 따라 만들어짐
  - 여러 개의 라운드로 이루어짐
  - 일반적으로 64bit 블록 크기, 16라운드를 사용

대칭 암호 원리

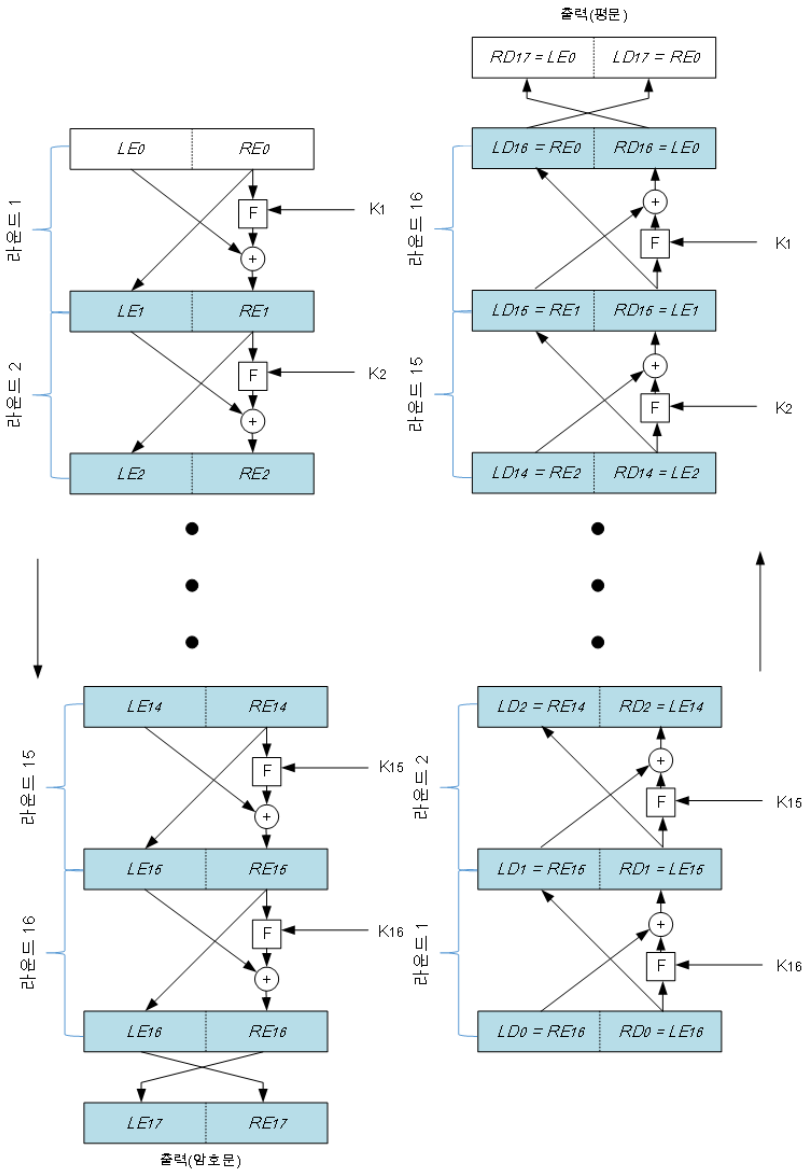
- Feistel 암호
  - 매개 변수와 설계 특성
    - 블록의 크기(Block size) : 크기가 클 수록 강한 보안 이지만 암호화/복호화 속도가 떨어짐
    - 키 크기(Key size) : 크기가 클 수록 강한 보안 이지만 암호화/복호화 속도가 떨어짐
    - 라운드수(Number of rounds) : 수를 증가시켜 보안을 강화할 수 있음
    - 서브키 생성 알고리즘(Subkey generation algorithm) : 복잡할수록 암호해독이 어려움
    - 라운드 함수(Round function) : 평문과 키를 입력 받는 함수로 복잡할수록 암호해독이 어려움

대칭 암호 원리

- Feistel 구조
  - 설계에 있어 고려할 두 가지 사항
    - 빠른 소프트웨어 암호/복호(Fast software encryption/decryption) : 알고리즘의 실행속도를 고려
    - 용이한 해독(Ease of analysis) : 알고리즘을 간결하고 명확하게 설명할 수 있으면 취약점을 찾기 쉬어 강한 보안성을 갖는 알고리즘을 만들수 있음
  - 복호 과정은 근본적으로 암호 과정과 동일

대칭 암호 원리

- Feistel 암호 구조
  - 평문 블록을 LE_0 과 RE_0 두 조각으로 나눔
  - $LE_i = RE_i$, $RE_i = LE_{i-1} \oplus f(RE_{i-1}, K_i)$
  - 위의 과정을 i라운드 만큼 실행
  - 마지막 결과 LE_{16} 과 RE_{16} 의 위치를 바꿈
  - 복호화 과정은 암호화 과정을 반대로 함(키는 K_{16}부터 역순으로)

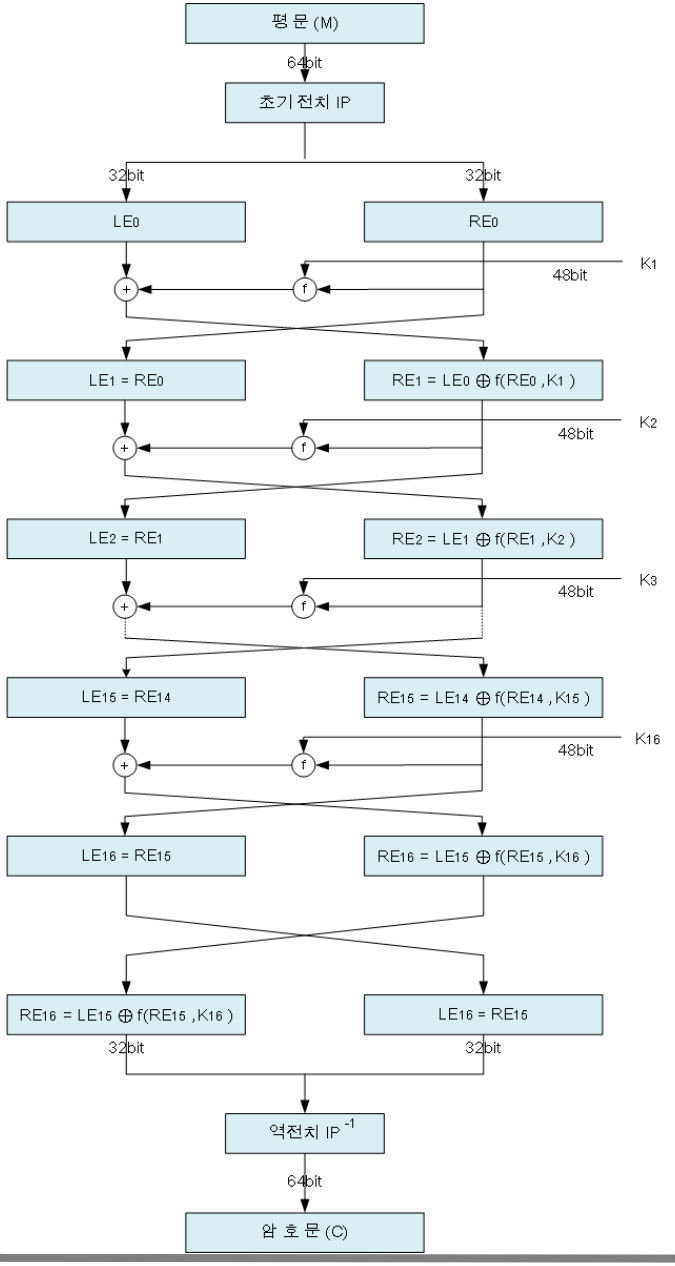


대칭 암호 알고리즘

- DES(Data Encryption Standard) 개요
  - 평문의 길이는 64bit이고 키의 길이는 56bit
  - Feistel 네트워크 변형된 형태
  - 라운드 회수는 16
  - 56bit 원래 키로부터 16개의 서브키를 생성
  - 8bit마다 패리티(Parity)비트 하나씩을 포함
  - DES 복호과정은 근본적으로 암호과정과 동일
  - 1998년 EFF의 “DES cracker”에 의해 암호가 깨짐

대칭 암호 알고리즘

- DES 암호화 과정
  - 초기 전치 IP를 거쳐 32bit씩 LE_0, RE_0으로 나누어짐
  - $LE_i = RE_{i-1}$, $RE_i = LE_{i-1} \oplus f(RE_{i-1}, K_i)$
  - 위의 과정이 16회 반복
  - 마지막 LE_i 과 RE_i의 위치를 바꿈
  - RE_{16} 과 LE_{16}은 초기전치의 역전치인 IP^{-1}을 거쳐 64bit 암호문이 됨



대칭 암호 알고리즘

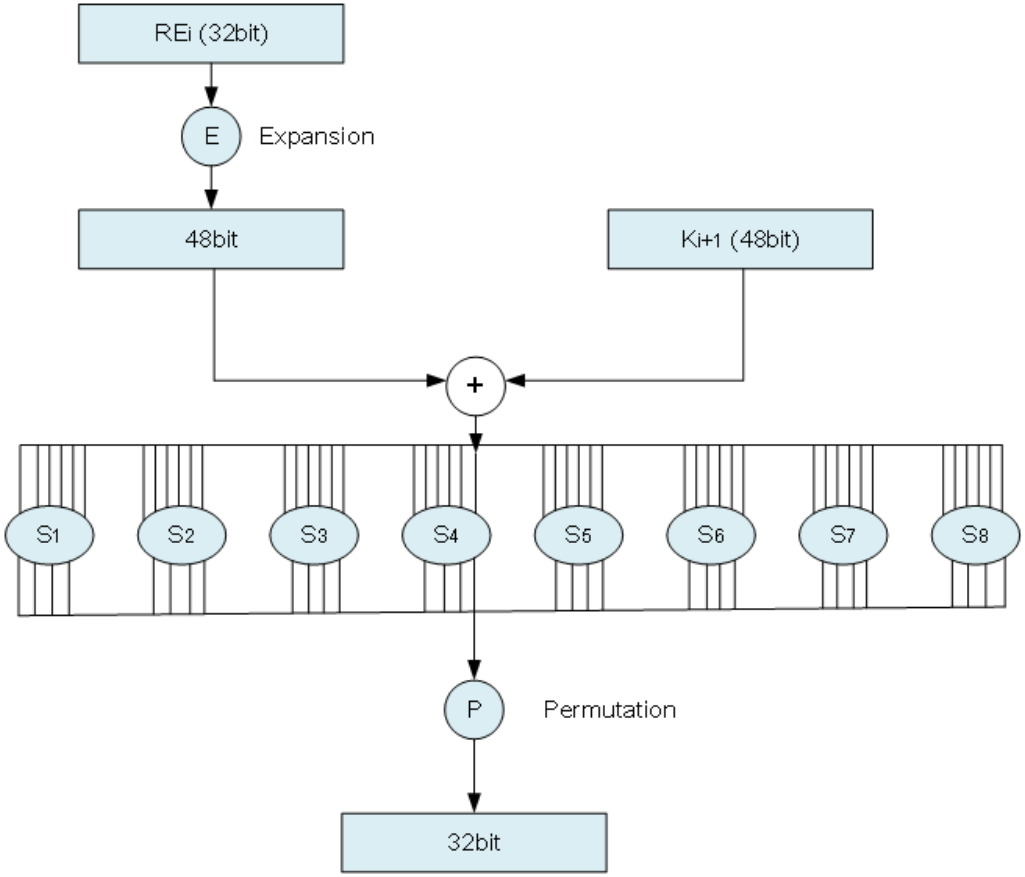
- DES 알고리즘 구조
  - 초기 전치(Initial permutation) IP
    - 64bit를 입력 받아 미리 정의된 규칙으로 재배열

IP 표							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IP^{-1}표							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

대칭 암호 알고리즘

- DES 알고리즘 구조
  - f함수 과정
    - REi 입력 32bit는 확대 전치(Expansion permutation)를 거쳐 48bit가 됨
    - E(REi)는 서브키 Ki+1과 XOR된 후 6bit씩 8개로 나누어져 8개의 S-Box에 입력됨
    - 각 S-Box의 출력이 4bit이므로 출력의 합은 32bit가 됨
    - 평형 전치(P-Box permutation)를 통해 f함수 출력



대칭 암호 알고리즘

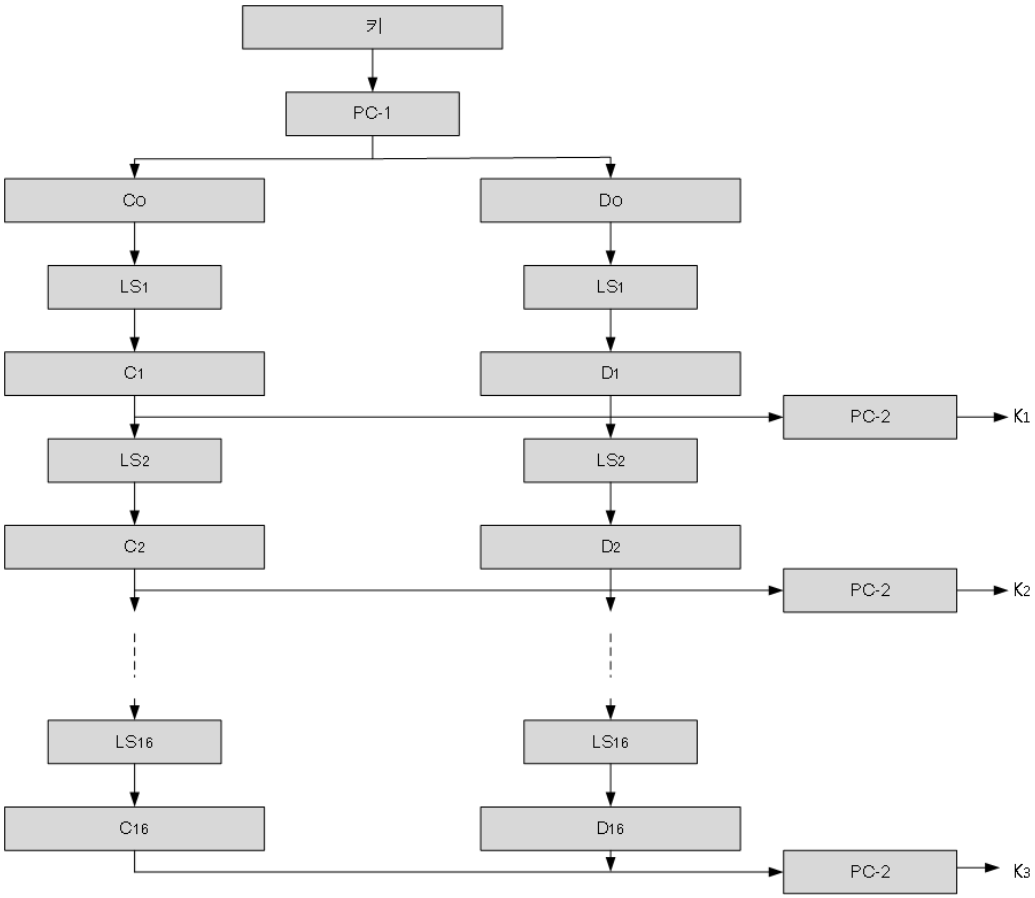
- DES 알고리즘 구조
  - f함수 과정
    - 확대 전치 E 와 평형 전치 P 표

확대 전치 E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

평형 전치 P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

대칭 암호 알고리즘

- DES 알고리즘 구조
  - 키 스케줄러(Key scheduler) 구조
    - 64bit 키에 8번째 비트마다 패리티 bit가 포함되어 실제 키 길이는 56bit
    - PC-1따라 전치 시킨 후 28bit씩 C_0,D_0으로 나뉨
    - C_i, D_i는 각각 LS_i에서 왼쪽으로 순환 시프트된 후 PC-2에 따라 56bit가 48bit로 축약 전치 됨



대칭 암호 알고리즘

- DES 알고리즘 구조
  - 키 스케줄러(Key scheduler) 구조
    - 키 스케줄러에 사용되는 표

키 전치 PC-1						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

축약 전치 PC-2					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

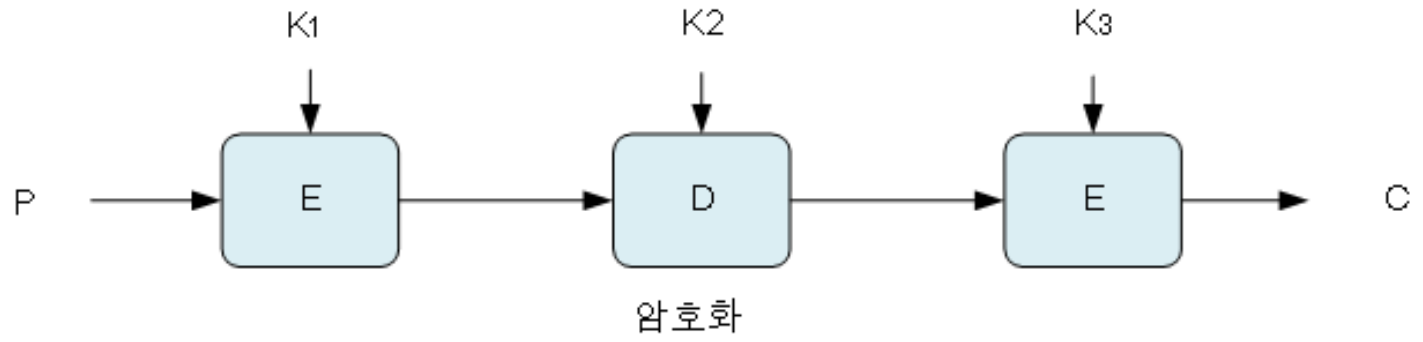
키 스케줄러 LS의 Shift 수			
위치	시프트	위치	시프트
LS_1	1	LS_9	1
LS_2	1	LS_{10}	2
LS_3	2	LS_{11}	2
LS_4	2	LS_{12}	2
LS_5	2	LS_{13}	2
LS_6	2	LS_{14}	2
LS_7	2	LS_{15}	2
LS_8	2	LS_{16}	1

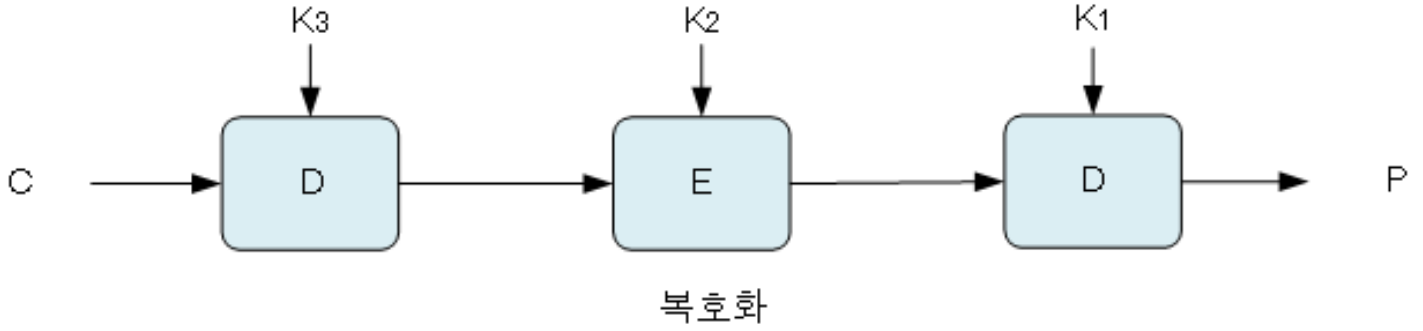
대칭 암호 알고리즘

- 3DES
  - DES 알고리즘을 세 번 수행
  - 각 56bit인 서로 다른 세 개의 키를 사용
  - 총 키의 길이가 168bit가 되어 DES의 취약점을 극복
  - 3DES의 주요 약점
    - 소프트웨어 구현 속도가 좀 느림
    - 64bit의 블록 크기를 사용하여 보안이나 효율성 면에서 떨어짐

대칭 암호 알고리즘

- 3DES
  - 3DES 그림
    - 암호-복호-암호 순서를 따름(C=E(K3, D(K2, E(K1, P))))



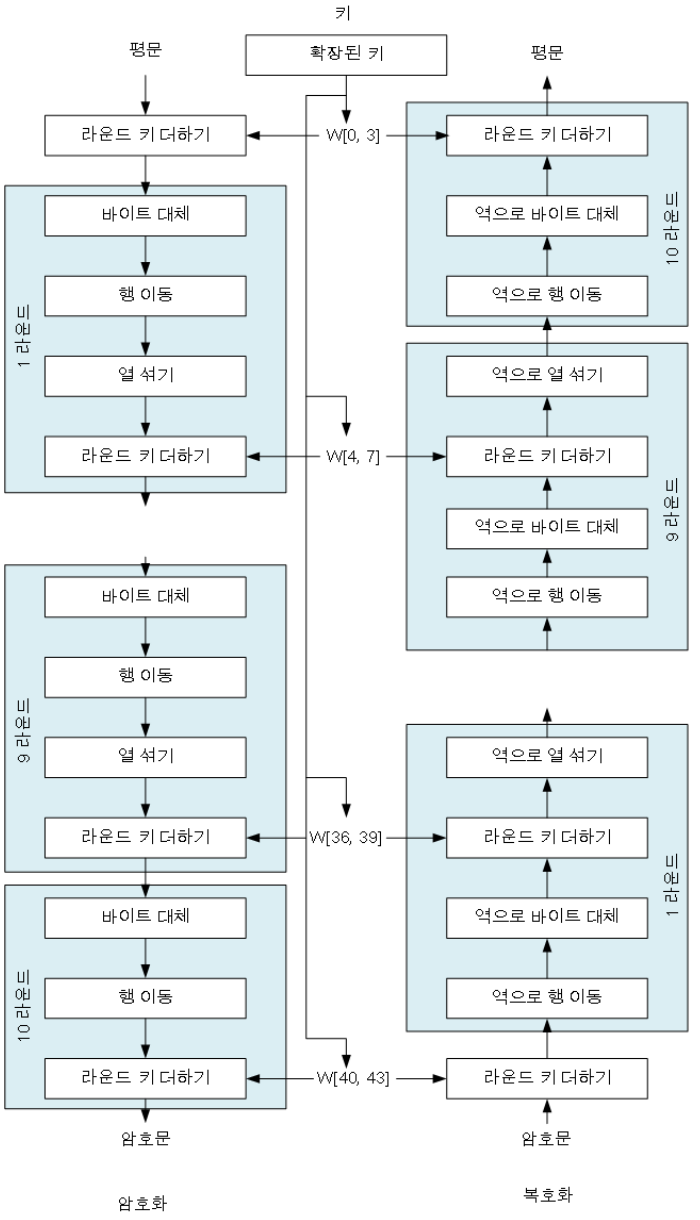


대칭 암호 알고리즘

- AES(Advanced Encryption Standard) 개요
  - 128bit 블록 크기와 128, 192, 또는 256bit의 키를 사용
  - 128bit 키 길이를 많이 사용
  - 암호와 복호를 할 때마다 블록을 상태 배열(State array)에 복사
  - 키를 키 스케줄 워드(Key schedule words)로 확장함
  - Feistel 구조가 아님

대칭 암호 알고리즘

- AES 알고리즘 구조
  - 총 10라운드로 구성
  - 시작과 끝은 라운드 키 더하기 단계
  - 9라운드까지 4단계로 구성
  - 마지막 라운드는 3단계로 구성
  - 라운드 키 더하기 단계에서만 키를 사용



대칭 암호 알고리즘

- AES 알고리즘 구조
  - 라운드 구조
    - 바이트 대체(Substitute bytes) : S-Box라는 표를 이용하여 바이트 단위 형태로 블록을 교환

상태 배열			
19	a0	9a	ea
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

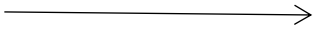
교환 후			
d4	e0	b8	1e
27	bf	b4	41
11	98	5d	52
ae	f1	e5	30

hex	S- Box															
	y															
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x 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	lf	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	08	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

대칭 암호 알고리즘

- AES 알고리즘 구조
  - 라운드 구조
    - 행 이동(Shift rows) : 행과 행을 치환
      - 첫 번째 행은 치환하지 않는다
      - 두 번째 행은 왼쪽으로 1칸씩 이동
      - 세 번째 행은 왼쪽으로 2칸씩 이동
      - 네 번째 행은 왼쪽으로 3칸씩 이동

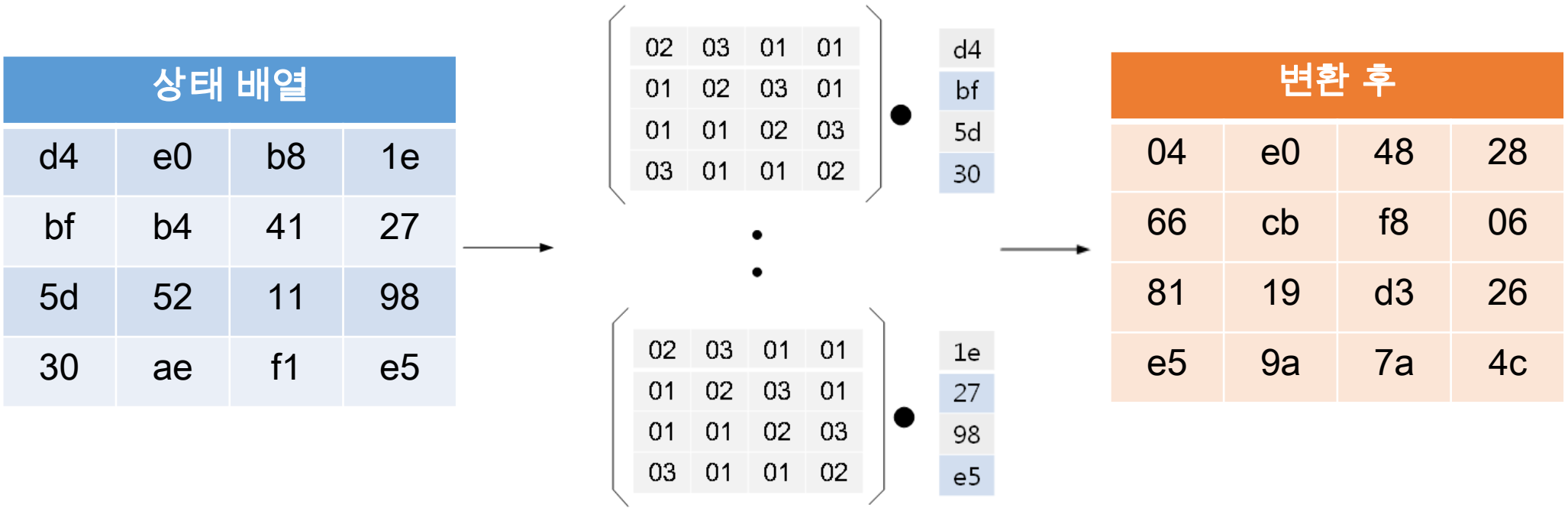
행 이동			
d4	e0	b8	1e
27	bf	b4	41
11	98	5d	52
ae	f1	e5	30



이동 후			
d4	e0	b8	1e
bf	b4	41	27
5d	52	11	98
30	ae	f1	e5

대칭 암호 알고리즘

- AES 알고리즘 구조
  - 라운드 구조
    - 열 섞기(Mix columns) : 열에 있는 각 바이트를 대체하여 변환

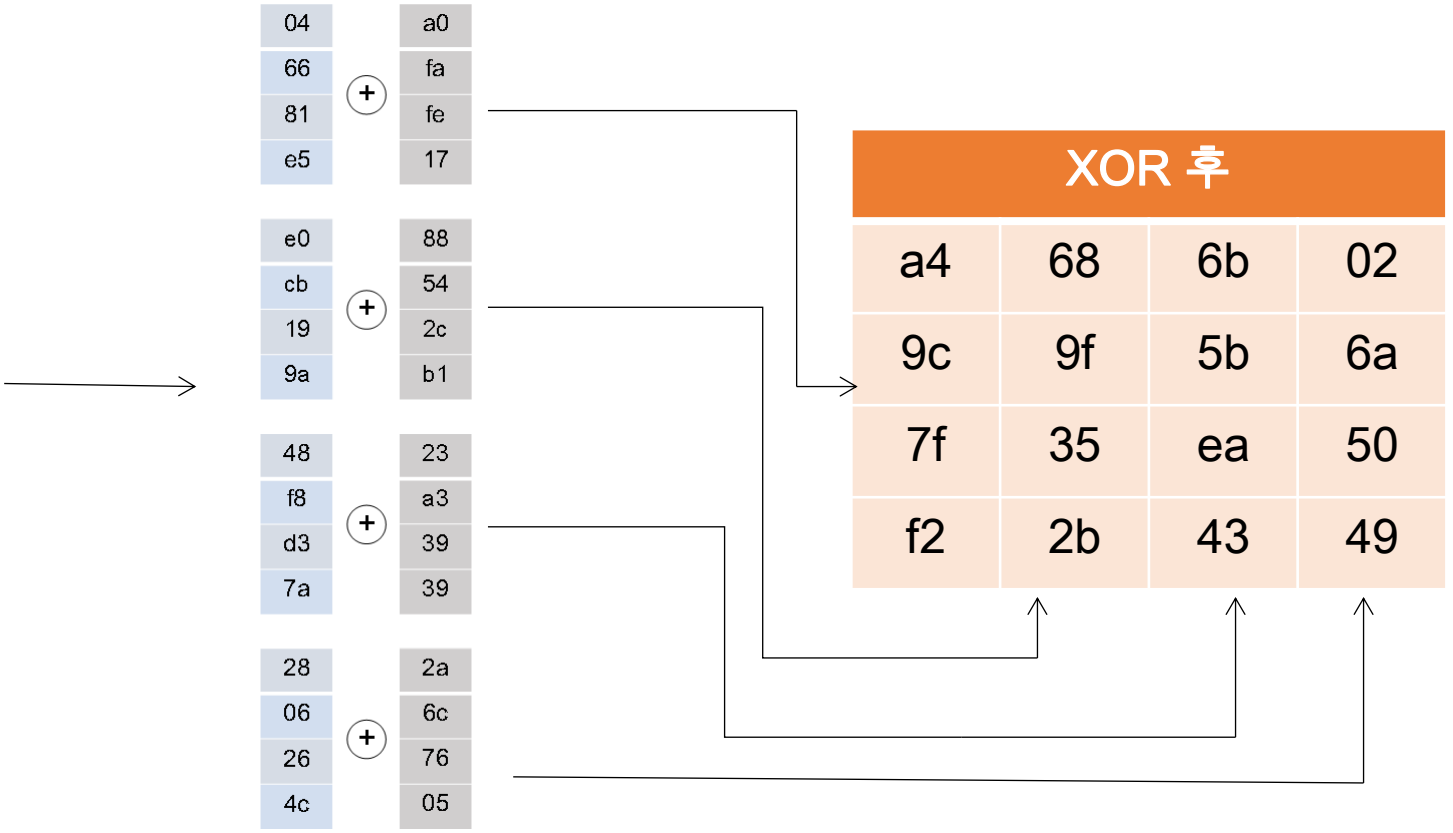


대칭 암호 알고리즘

- AES 알고리즘 구조
  - 라운드 구조
    - 라운드 키 더하기(Add round key) : 확장된 키의 일부와 현재 블록을 비트 별로 XOR함

상태 배열			
04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	4c

확장된 키			
a0	88	23	2a
fa	54	a3	6c
fe	2c	39	76
17	b1	39	05



대칭 암호 알고리즘

- AES 알고리즘 구조
  - 키 확장(Key expansion) 과정
    - 키 상태배열에 맨 오른쪽 열의 순서를 한 칸씩 이동함
    - S-Box를 적용

암호 키			
2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

09	cf
cf	4f
4f	3c
3c	09

8a
84
eb
01

S-Box

hex	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
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

x

대칭 암호 알고리즘

- AES 알고리즘 구조
  - 키 확장(Key expansion) 과정
    - 첫 번째 열, S-Box로 대체한 열과 준비된 테이블(Rcon) 맨 왼쪽 열을 XOR 연산
    - 연산 결과를 새로운 행렬 맨 왼쪽 열에 넣음
    - 새로운 라운드 키 행렬의 맨 왼쪽 열 부분에만 Rcon을 사용

암호 키				라운드 키 1			
2b	28	ab	09	a0			
7e	ae	f7	cf	fa			
15	d2	15	4f	fe			
16	a6	88	3c	17			

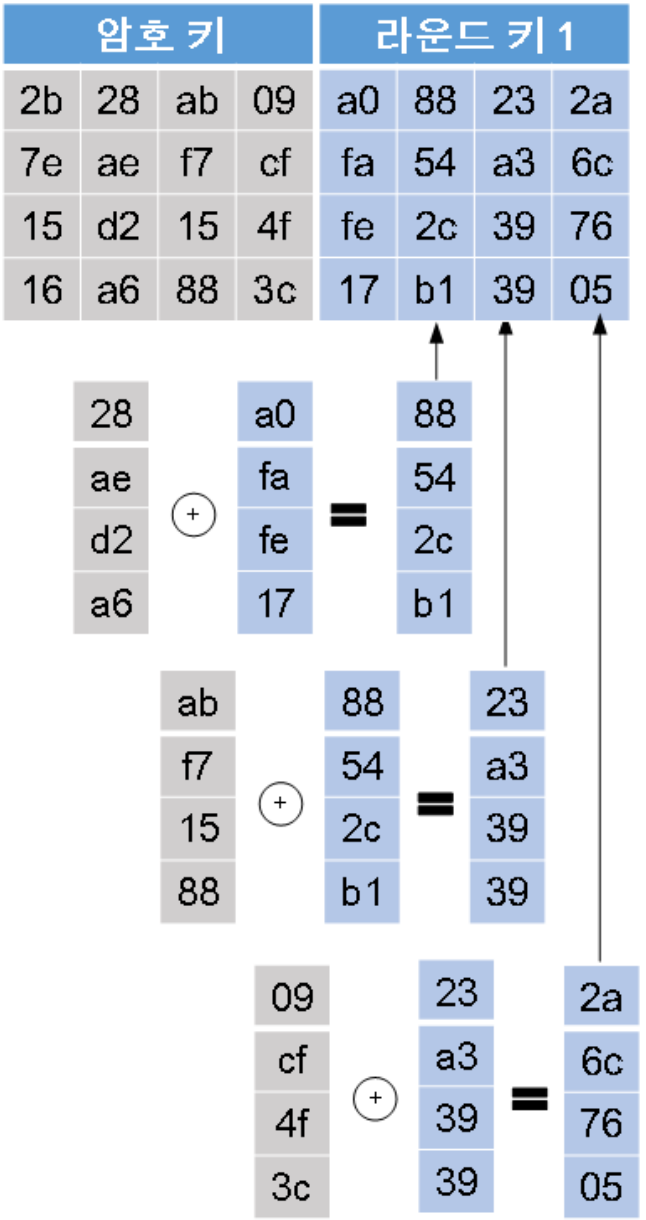
2b	8a	01	=	a0
7e	84	00		fa
15	eb	00		fe
16	01	00		17

Rcon(4)

Rcon									
01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

대칭 암호 알고리즘

- AES 알고리즘 구조
  - 키 확장(Key expansion) 과정
    - 두 번째 열 과 라운드 키 맨 왼쪽 부터 XOR연산 후 채움
    - 이 과정을 반복해서 라운드 키를 생성



랜덤넘버와 의사랜덤넘버

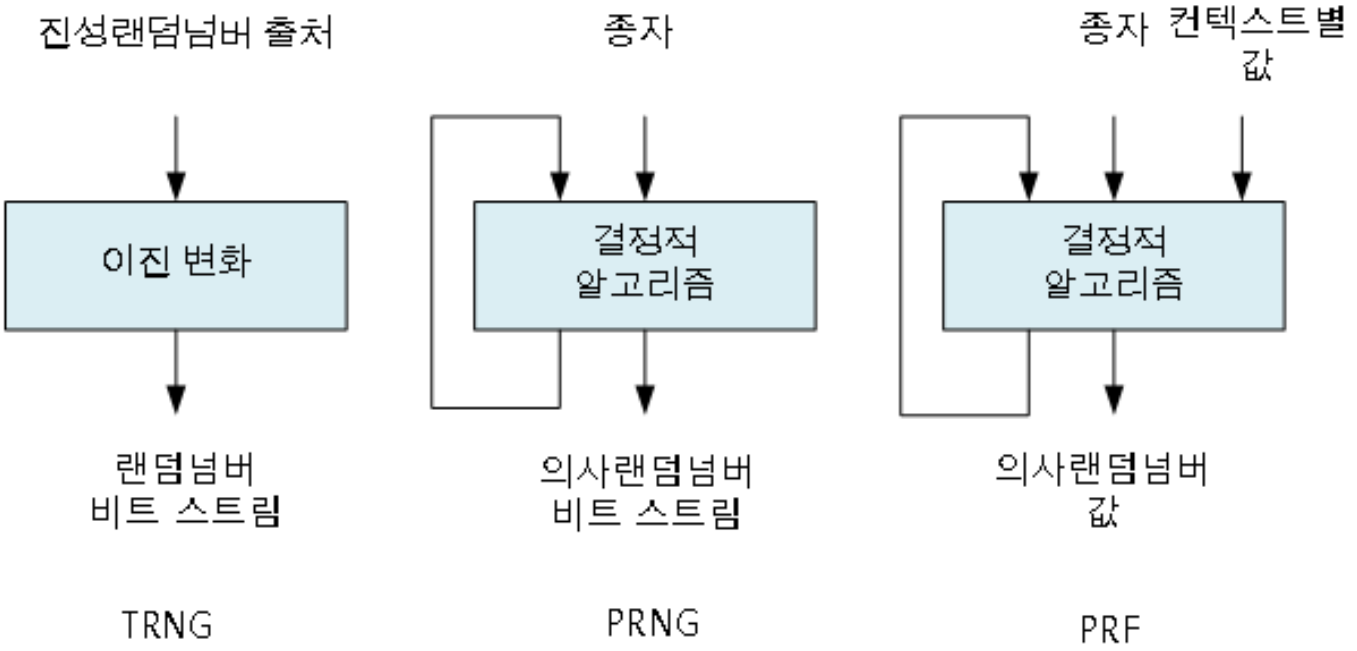
- 랜덤넘버(Random number)
  - 무작위성(Randomness)
    - 무작위성의 두 가지 기준
      - 균등분포(Uniform distribution) : 수열의 비트분포가 반드시 균등해야 함
      - 독립성(Independence) : 어떠한 부분수열도 다른 수열로부터 추론할 수 없어야 함
  - 예측불가능성(Unpredictability)
    - 수열의 일부를 보고 다음에 이어지는 수를 예측할 수 없어야 함
- 의사랜덤넘버(Pseudorandom number)
  - 알고리즘의 상태에 의해 값이 정해지므로 수열은 일정한 주기를 가짐

랜덤넘버와 의사랜덤넘버

- TRNG, PRNG와 PRF
  - 진성랜덤넘버 생성기(True Random Number Generator)
    - 엔트로피 소스(Entropy sourece) 사용
      - 컴퓨터의 물리적 환경에서 얻을 수 있는 값
  - 의사랜덤넘버 생성기(Pseudorandom Number Generator)
    - 종자(seed)를 사용
      - 고정된 값
    - 입력 값이 같으면 출력 값이 같음
    - 피드백 경로가 있음
      - 출력 값을 다시 입력 값으로 사용하기도 함
    - 무한 비트열을 생성 함

랜덤넘버와 의사랜덤넘버

- TRNG, PRNG와 PRF
  - 의사랜덤넘버 함수(Pseudorandom Function)
    - 고정된 길이 의사랜덤 비트열을 생성 하는 데 사용하는 함수
    - 생성되는 비트열만 다르지 PRNG와 차이점은 없음
  - TRNG, PRNG와 PRF 그림



랜덤넘버와 의사랜덤넘버

- 의사랜덤넘버(Pseudorandom number)
  - 알고리즘의 설계
    - 특정 목적 알고리즘
      - 의사랜덤 비트 스트림을 생성하기 위해서 특정한 목적만을 위해 설계된 알고리즘
    - 기존 암호 알고리즘을 이용한 알고리즘
      - 암호학적 알고리즘은 랜덤화된 입력 효과를 가짐
      - PRNG의 핵심 역할

감사합니다!