2018/01/10, 2018 보안 기초 세미나

Network Security Essentials

- Chapter_2 대칭 암호와 메시지 기밀성(1) -

권순홍 (soonhong@pel.smuc.ac.kr)

상명대학교 프로토콜공학연구실

목 차

- 대칭 암호 원리
- 대칭 암호 알고리즘
- 랜덤넘버와 의사랜덤넘버

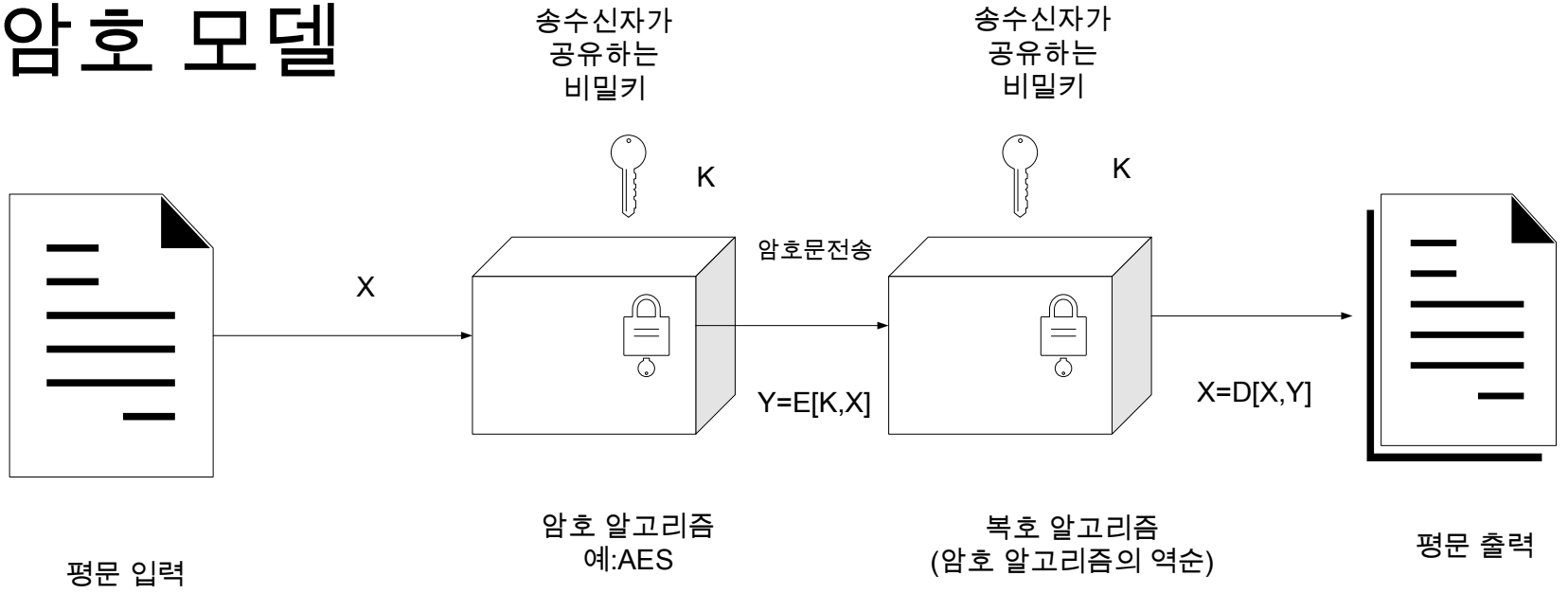
대칭 암호 원리

- 대칭 암호(Symmetric Encryption)구조
  - 평문(Plaintext)
    - 원문이나 데이터로써 알고리즘의 입력으로 이용
  - 암호 알고리즘(Encryption algorithm)
    - 원문을 다양한 방법으로 대체(Substitution)하고 치환(Permutation)하는 것
  - 대칭 키(Symmetric Key)
    - 메시지를 암호화할 때 사용하는 키
  - 암호문(Ciphertext)
    - 출력으로 나오는 암호화된 메시지
  - 복호 알고리즘(Decryption algorithm)
    - 암호 알고리즘을 역으로 수행하는 것

대칭 암호 원리

- 대칭 키 보안을 위한 필수사항
  - 강한 암호 알고리즘이 있어야함
  - 송신자와 수신자는 공유하는 대칭키를 안전한 방법으로 획득해야 하고 안전하게 보관해야 함

- 대칭 암호 모델



대칭 암호 원리

- 암호시스템
  - 3가지 독립적 단계
    1. 평문을 암호문으로 전환하는 데 사용되는 연산 유형
    2. 사용되는 키
    3. 평문의 처리되는 방법

대칭 암호 원리

- 3가지 독립적인 단계
  1. 평문을 암호문으로 전환하는 데 사용되는 연산 유형
     - 대체(Substitution)
       - 평문의 각 요소(비트,문자,블록)를 다른 요소로 바꾸는 것
     - 치환(Permutation)
       - 요소의 순서(위치)를 재조정하는 것

대칭 암호 원리

- 3가지 독립적인 단계

2. 사용되는 키
   - 대칭키
     - 송신자, 수신자 양측이 동일한 키를 사용
     - 비밀키로 암호화한 것을 같은 비밀키로 복호화

대칭 암호 원리

- 암호 해독 및 메시지 공격 유형
  - 암호해독(Cryptanalysis)
    - 평문이나 키를 찾으려는 시도
  - 암호화된 메시지 공격 유형
    - 전수 공격(Brute-force Attack)
    - 암호문만 알고 있는 공격(Ciphertext-only Attack)
    - 알려진 평문 공격(Known-Plaintext Attack)
    - 선택 평문 공격(Chosen-Plaintext Attack)
    - 선택 암호문 공격(Chosen-Ciphertext Attack)

대칭 암호 원리

- 암호화된 메시지 공격 유형
  - 암호문만 알고 있는 공격(Ciphertext-only Attack)
    - 통계적 성질과 문자의 특성 등을 추정하여 해독
      - 전수공격(Brute Force Attack)
        - 가능한 모든 경우의 수를 시도해보는 공격
      - 통계적 분석 공격(Statistical Attack)
        - 암호문에서 통계적으로 많이 사용되는 평문 언어의 고유한 특징으로부터 정보를 얻는 공격
      - 패턴 공격(Pattern Attack)
        - 암호문에 존재하는 패턴을 이용하여 평문을 유추하는 공격

대칭 암호 원리

- 암호화된 메시지 공격 유형
  - 알려진 평문 공격(Known Plaintext Attack)
    - 평문, 암호문을 알고 있는 상태에서 암호문과 평문의 연관성을 유추하여 암호를 해독하는 공격 유형
      - e.g., 전수공격, 통계적 분석, 패턴 공격

대칭 암호 원리

- 암호화된 메시지 공격 유형(수정)
  - 선택 평문 공격(Chosen Plaintext Attack)
    - 공격자가 송신자의 시스템 근원지에 접속하여 평문을 선택하고 그 평문에 해당하는 암호문을 얻어 암호를 해독하는 공격
  - 선택 암호문 공격(Chosen Ciphertext Attack)
    - 공격자가 수신자의 컴퓨터에 접속하여 암호문을 선택하고 그 암호문에 해당하는 평문을 얻어 암호를 해독하는 공격
  - 선택문 공격(Chosen Attack)
    - 선택평문과 선택 암호문 공격을 합친 공격

대칭 암호 원리

- 키 탐색 시간에서의 안전한 구조
  - 암호문을 깨는데 드는 비용이 암호화된 정보의 가치보다 큼
  - 암호문을 깨는데 걸리는 시간이 해당 정보의 수명보다 김

  - 키 탐색에 요구되는 평균 시간(전수공격일 경우)

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 구조
  - 1973년 IBM의 Horst Feistel이 최초로 발표한 구조를 따라 만들어짐
  - 대부분의 대칭 블록 암호 알고리즘의 구조는 Feistel 암호 구조를 따라 만들어짐
    - e.g., DES,3DES
  - 여러 개의 라운드로 이루어짐
    - 기본적으로 16라운드를 사용, 사용자가 라운드수 조정 가능
  - 보통 64비트 블록 크기

대칭 암호 원리

- Feistel 암호 구조
  - 매개 변수와 설계 특성
    - 블록의 크기(Block size)
      - 크기가 클 수록 강한 보안이나 암호화/복호화 속도가 떨어짐
    - 키 크기(Key size)
      - 크기가 클 수록 강한 보안이나 암호화/복호화 속도가 떨어짐
    - 라운드 수(Number of rounds)
      - 수를 증가시켜 보안을 강화 할수 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 암호 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 강한 보안성을 갖는 알고리즘을 생성 가능
  - 복화 과정은 근본적으로 암호 과정과 동일

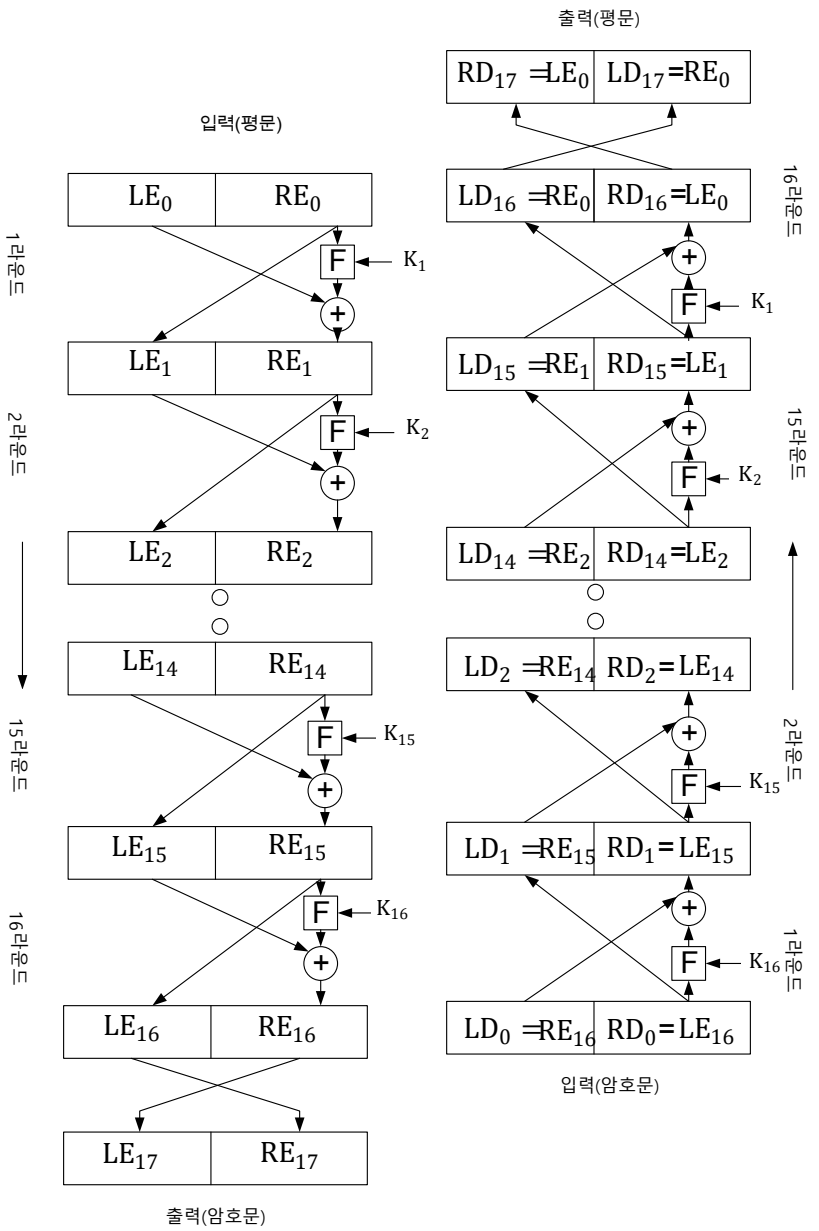
대칭 암호 원리

- Feistel 암호 구조

1. 평문 블록을 LE_0와 RE_0 두 조각으로 나눔
2. $LE_i = RE_{i-1}$, $RE_i = LE_{i-1} \oplus f(RE_{i-1}, K_i)$
3. 1,2번 i라운드 만큼 실행
4. 마지막 결과 LE_{16} 과 RE_{16} 의 위치를 바꿈
5. 복호화 과정은 암호화 과정의 역순으로 실행

대칭 암호 원리

- Feistel 암호 구조(수정)
  - 입력(평문)
    - 64비트
  - LE_i, RE_i
    - 32비트

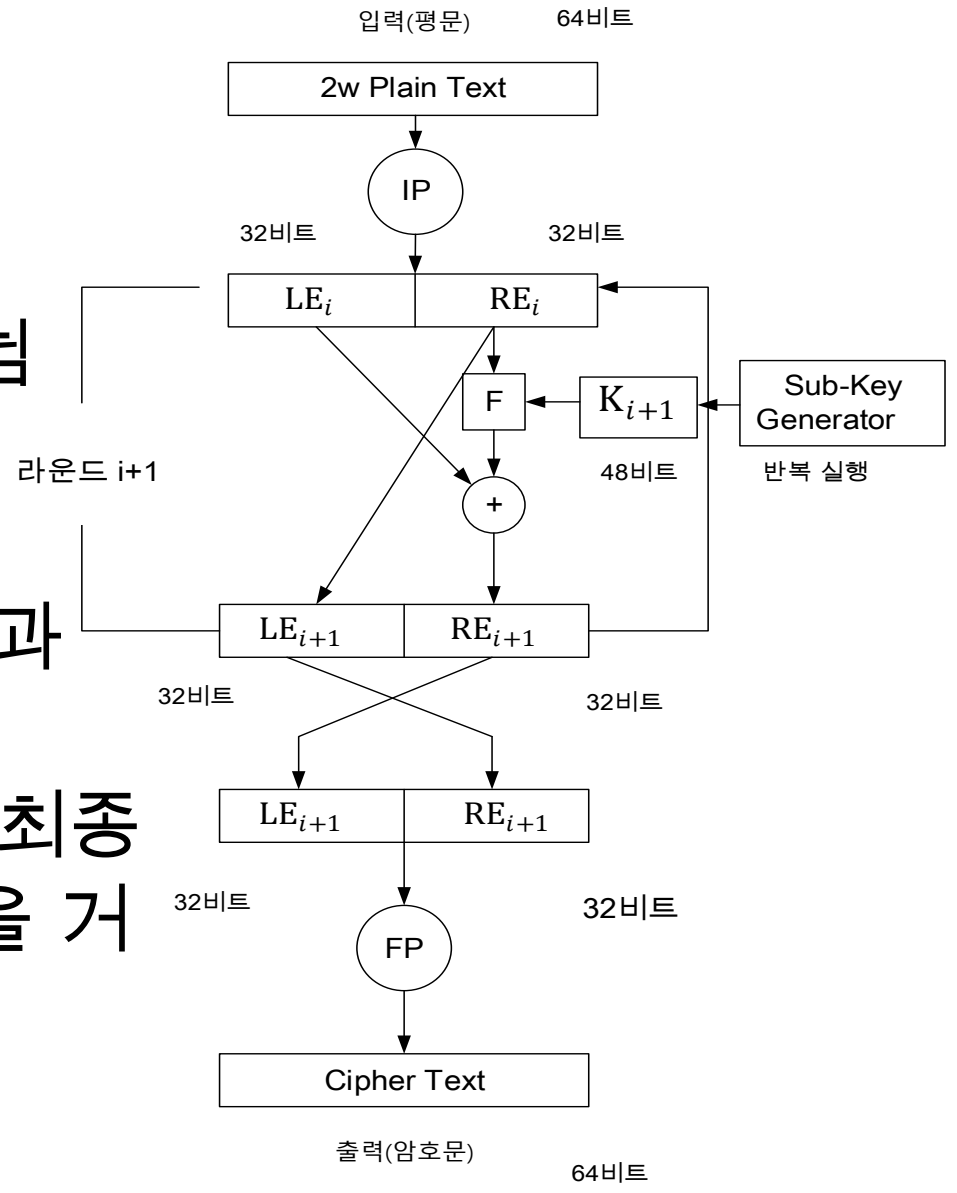


대칭 암호 알고리즘

- DES(Data Encryption Standard) 개요
  - 1972년 미국 국가기술표준원(NIST)이 개발한 미국 정부 규모의 표준적인 암호 알고리즘
- 특징
  - 평문의 길이는 64비트이고 키의 길이는 56비트
    - 이보다 긴 평문은 64비트 블록으로 나눔
  - Feistel 암호 구조 변형된 형태
  - 라운드 회수는 16
  - 56비트 원래 키로부터 16개의 서브키를 생성
  - DES 복호 과정은 근본적으로 암호과정과 동일
  - 1998년 EFF(Electronic Frontier Foundation)의 “DES Cracker”에 의해 암호가 깨짐

대칭 암호 알고리즘

- DES 암호화 과정(수정)
  - 입력으로 들어온 64비트 평문 블록은 초기 치환 IP를 거쳐 LE_i와 RE_i으로 32비트씩 나뉨
  - $RE_i = LE_{i+1}$
  - $F(RE_i, K_{i+1}) \oplus LE_i = RE_{i+1}$
  - 마지막 라운드가 끝나면 LE_{16}과 RE_{16}교환
  - 위치가 교환된 평문 블록들은 최종 치환(초기 치환의 역치환)FP을 거쳐 64비트 암호문으로 출력됨



대칭 암호 알고리즘

- DES 알고리즘 구조(수정)
  - 초기 전치(Initial Permutation)
    - 64비트를 입력 받아 정의된 규칙에 따라 재배열
  - 최종 치환(Final Permutation)
    - 초기 치환의 역 관계

초기 치환							
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

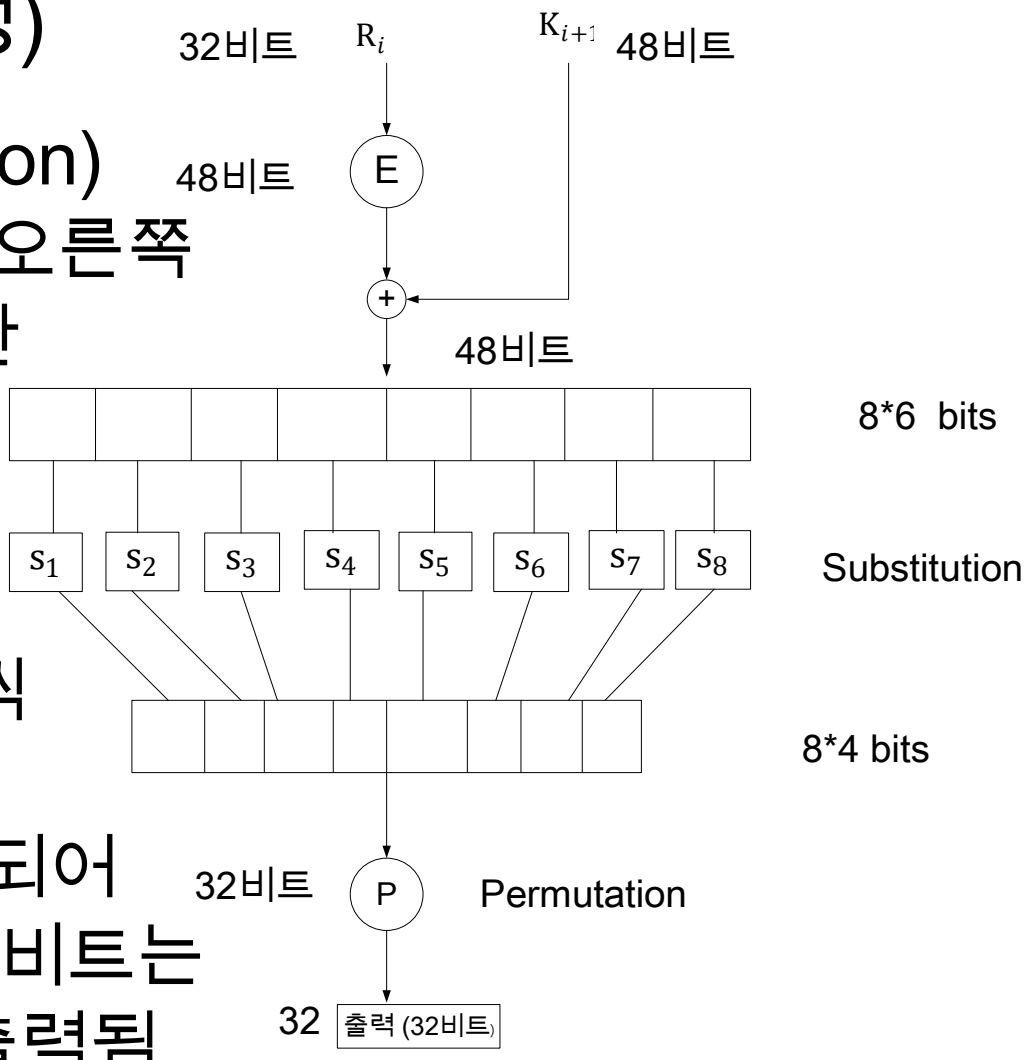
최종 치환							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

대칭 암호 알고리즘

- DES 알고리즘 구조
  - 라운드 함수
    - S-Box(Substitution-Box)
      - 6비트의 입력을 4비트의 출력으로 축소시켜 변환하는 함수
      - 역방향으로 복원이 어려움
        - 역방향 복원은 Inverse S-Box를 통해 복원이 가능
      - 1비트 입력이 2비트 이상의 출력을 나타내어야 함
    - P-Box(Permutation-Box)
      - S-Box 입력을 통해 비트를 치환하여 출력하는 함수

대칭 암호 알고리즘

- DES 알고리즘 구조(수정)
  - 라운드 함수(Round Function)
    - 입력으로 들어온 32비트 오른쪽 평문 블록 RE_i은 확장치환 E를 거쳐 48비트가 됨
    - 48비트로 확대된 RE_i은 서브키 K_{i+1}와 XOR연산 후 8개의 S-Box에 6비트씩 입력됨
    - S-Box에서 4비트씩 축소되어 출력된 비트들의 총 합 32비트는 P-Box를 통해 함수에서 출력됨



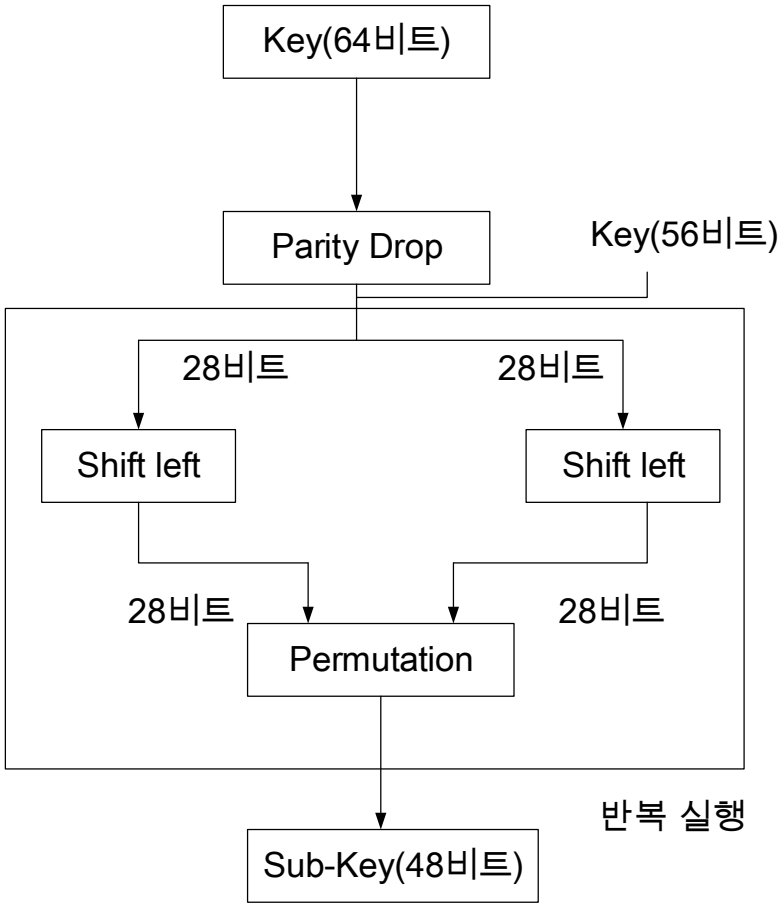
대칭 암호 알고리즘

- DES 알고리즘 구조(수정)
  - 라운드 함수
    - 확장 치환(Expansion Permutation)
      - 32비트를 입력 받아 정의된 규칙에 따라 48비트로 확장

확장 치환					
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

대칭 암호 알고리즘

- DES 알고리즘 구조(그림 수정)
  - 서브키 생성기(Sub-Key Generator)
    - 입력으로 들어온 64비트 키는 Parity Drop을 거쳐 키 생성 과정 전에 Parity bits(8bits)를제거
    - 56비트 키를 28비트씩 나눠 비트를 좌측으로 순호나 이동시킴
      - 1,2,9,16 라운드는 1비트씩
      - 나머지 라운드 2비트씩
    - 축소 치환을 통해 56비트 키를 48비트로 축소하여 서브키로 출력
    - 위 과정 16회 반복 실행



대칭 암호 알고리즘

- DES 알고리즘 구조
  - 키 스케줄러 구조

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

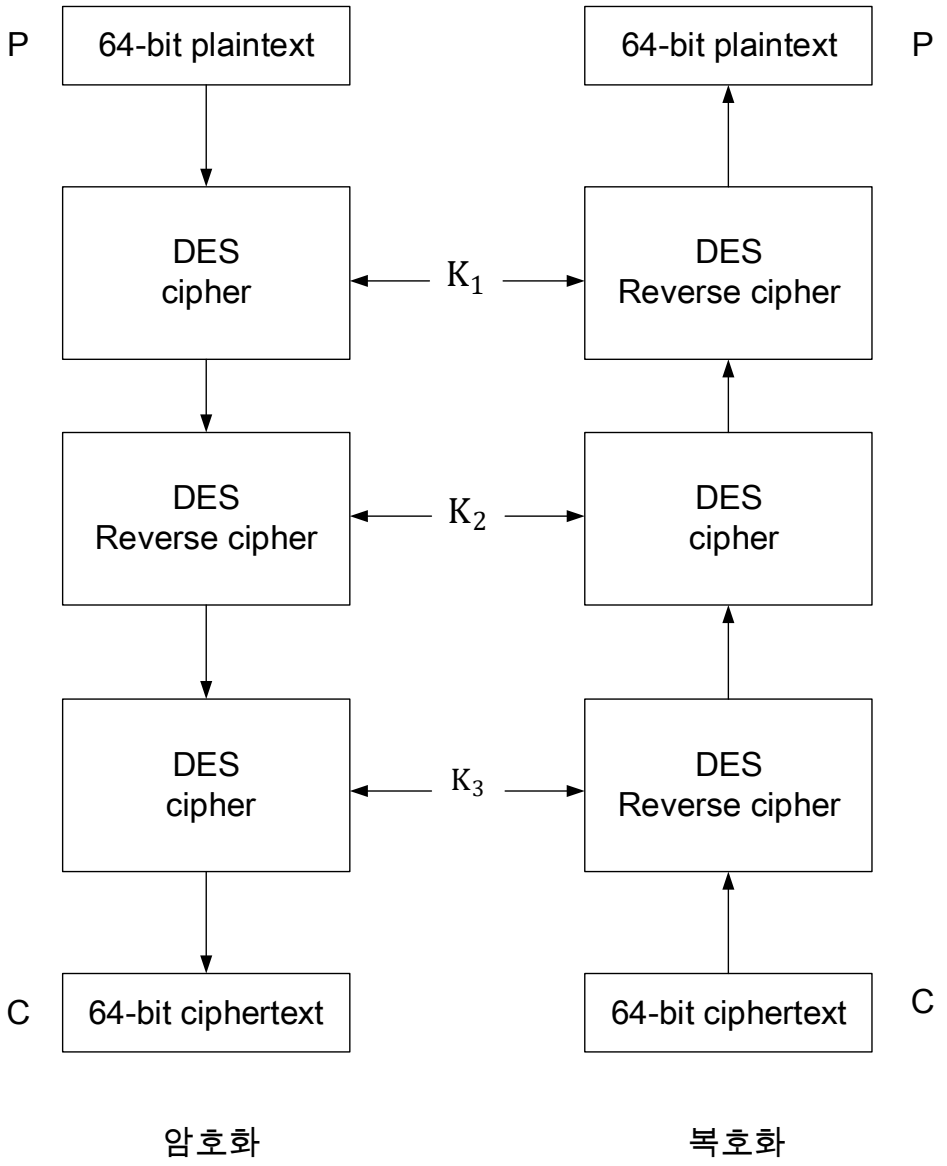
키 스케줄러 LS의 Shift 수			
위치	시프트	위치	시프트
LS_1	1	LS_9	1
LS_2	1	LS_{10}	2
LS_3	2	LS_{11}	2
LS_4	2	LS_{12}	2
LS_5	2	LS_{13}	2
LS_6	2	LS_{14}	2
LS_7	2	LS_{15}	2
LS_8	2	LS_{16}	1

대칭 암호 알고리즘

- 3DES
  - DES 알고리즘을 세 번 수행
    - 알고리즘 소프트웨어적으로 느림
  - 각 56비트인 서로 다른 세 개의 키를 사용
  - 64 비트블록 사용
    - 보안이나 효율성면에서 떨어짐
  - 총 키의 길이가 168비트가 되어 DES의 취약점을 극복
    - DES알고리즘을 3번 반복함

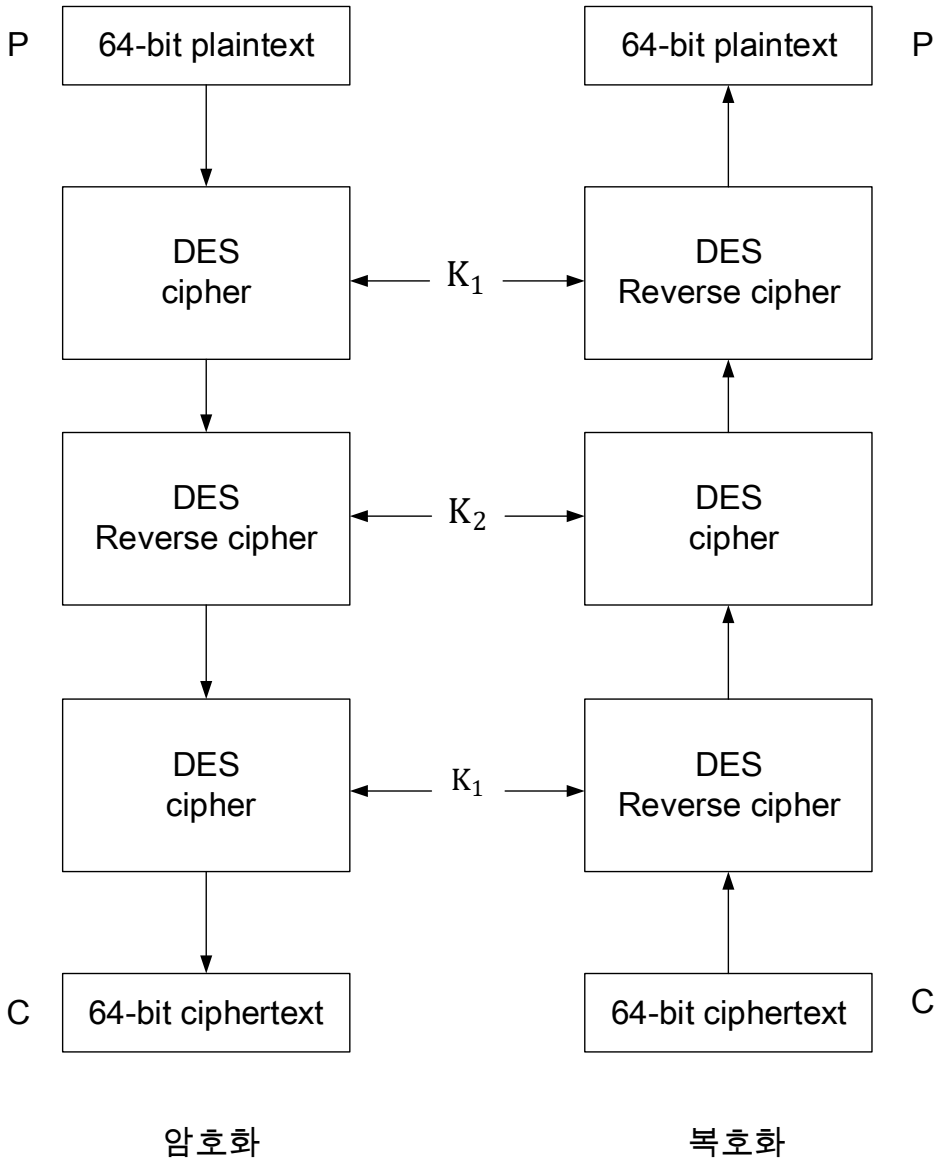
대칭 암호 알고리즘

- 3DES(그림 수정)
  - 3개의 키를 갖는 삼중DES



대칭 암호 알고리즘

- 3DES(그림 수정)
  - 2개의 키를 갖는 삼중DES



대칭 암호 알고리즘

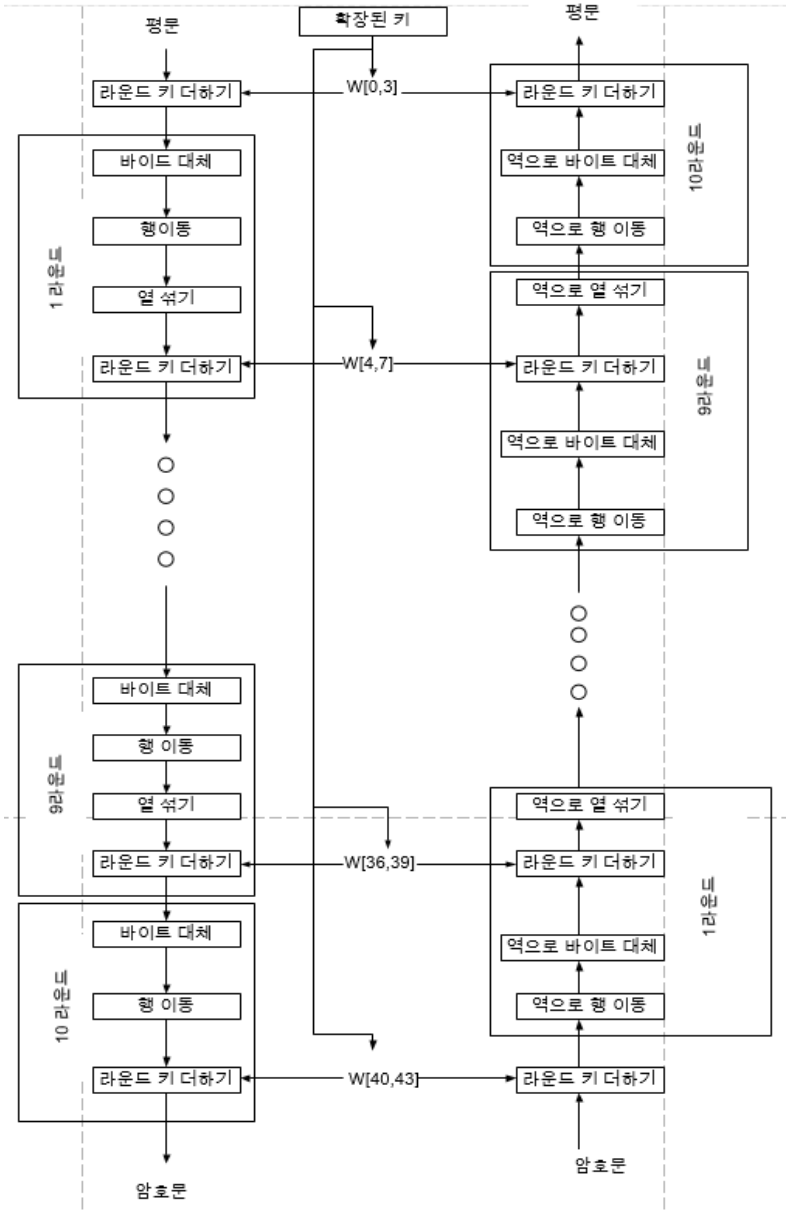
- AES(Advanced Encryption Standard)
  - 2001년 미국 국립기술표준원(NIST)에서 공표한 대칭키 암호 알고리즘
  - 두 명의 벨기에 암호학자 Joan Deamen과 Vincent Rijmen이 개발한 Rijndael 블록 암호 알고리즘
  - 특징
    - 128비트 블록과 길이가 128, 192, 256비트인 키를 사용
      - 128비트- 10라운드, 192비트-12 라운드, 256비트-14라운드
    - 기본적인 SPN(Substitution-Permutation Network) 구조 블록 암호 알고리즘 특성을 가짐
      - S-box, P-box를 이용하여 비트 이동 없이 한 번에 암호화 ,복호화 함
    - 복호화 과정은 암호화 과정의 역순

대칭 암호 알고리즘

- AES 알고리즘 구조
  - 총 10라운드로 구성
  - 시작과 끝은 라운드 키 더하기 단계
  - 9라운드까지 4단계로 구성
  - 마지막 라운드는 3단계로 구성
    - AES 암호가 역으로 작동되기 위해 필요한 구조적 성질
  - 라운드 키 더하기 단계에서만 키를 사용

대칭 암호 알고리즘

- AES 알고리즘 구조
  - 순서
    1. 평문과 라운드키를 XOR 함
    2. 바이트를 치환
    3. 행렬로 바이트를 옮김
    4. 열 별로 바이트를 섞음
    5. 라운드키와 XOR함
    6. (라운드 수-1)만큼 2~5반복
    7. 마지막 라운드는 열 섞기를 제외하고 수행



대칭 암호 알고리즘

- AES 알고리즘 구조
  - 라운드 구조
    - 바이트 대체(Substitute bytes)
      - S-box라는 표를 이용하여 바이트 단위 형태로 블록을 교환
        - e.g., 0x19을 치환하면 0xd4로 바뀐다.

19	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

변환

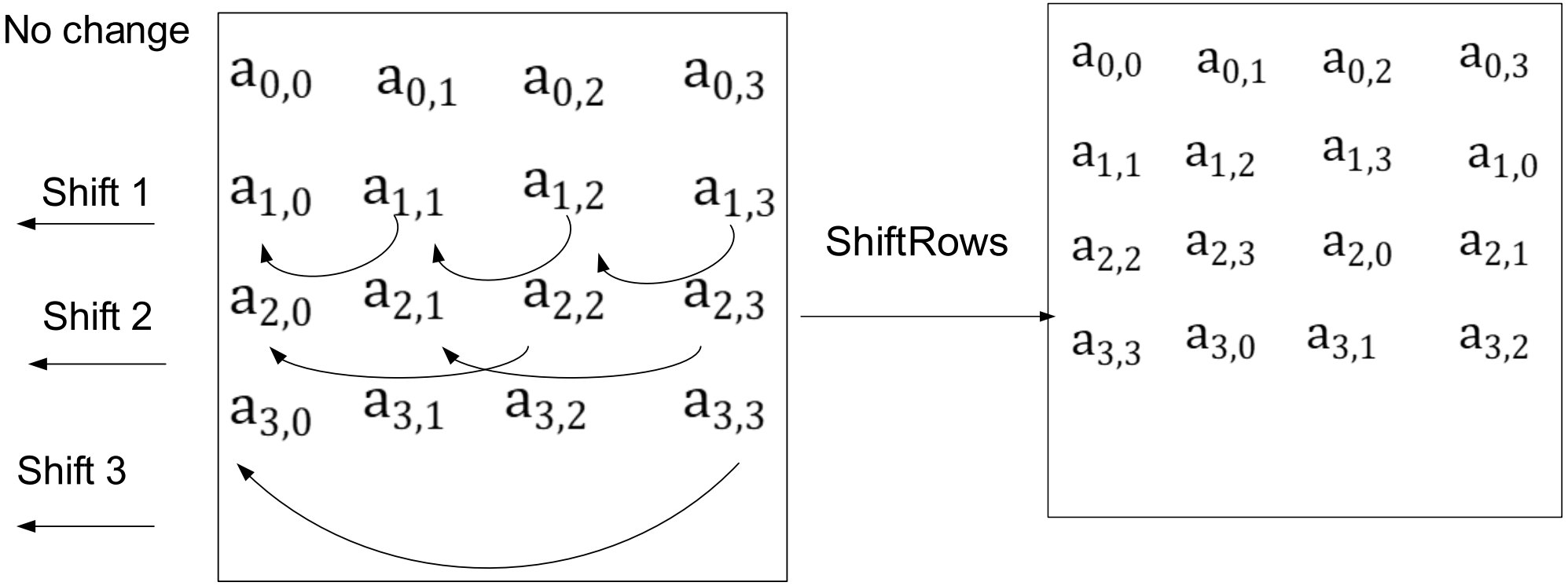
d4	e0	b8	1e
27	bf	b4	41
11	98	5d	52
ae	f1	e5	30

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

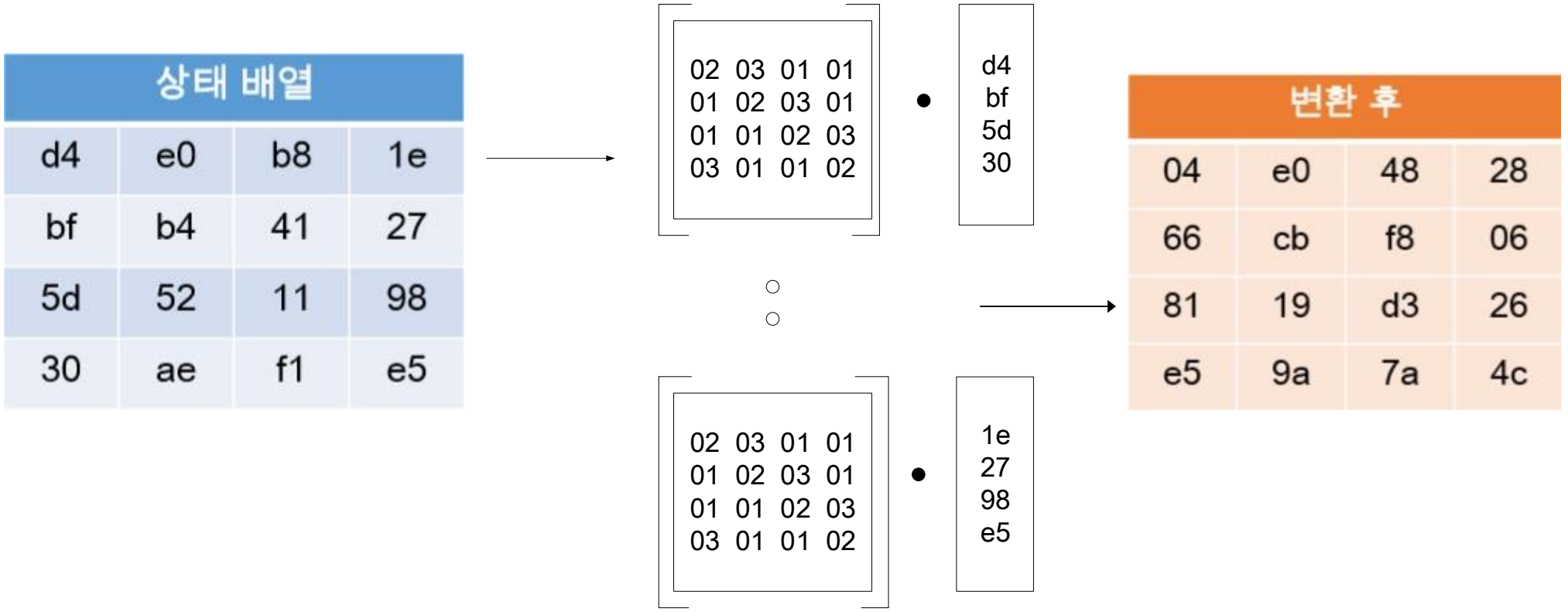
대칭 암호 알고리즘

- AES 알고리즘 구조
  - 라운드 구조
    - 행 이동(Shift rows):행과 행을 치환



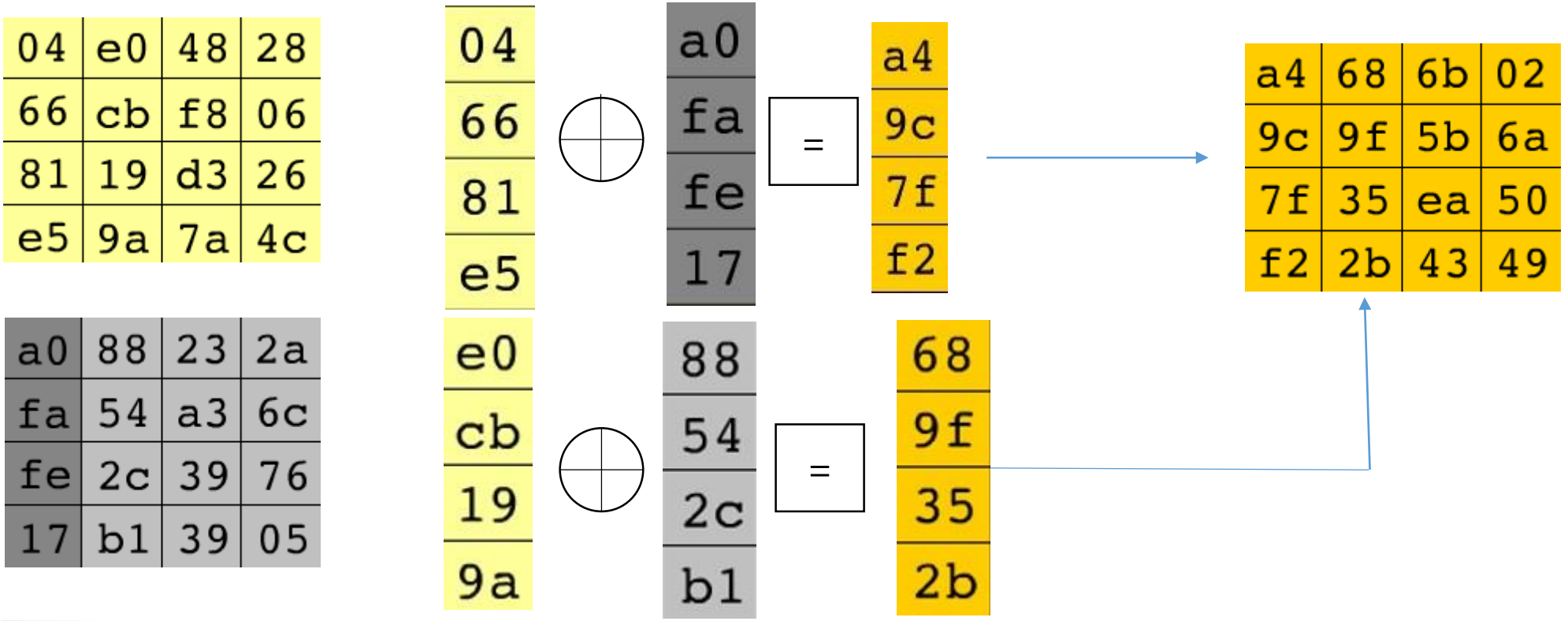
대칭 암호 알고리즘

- 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



대칭 암호 알고리즘

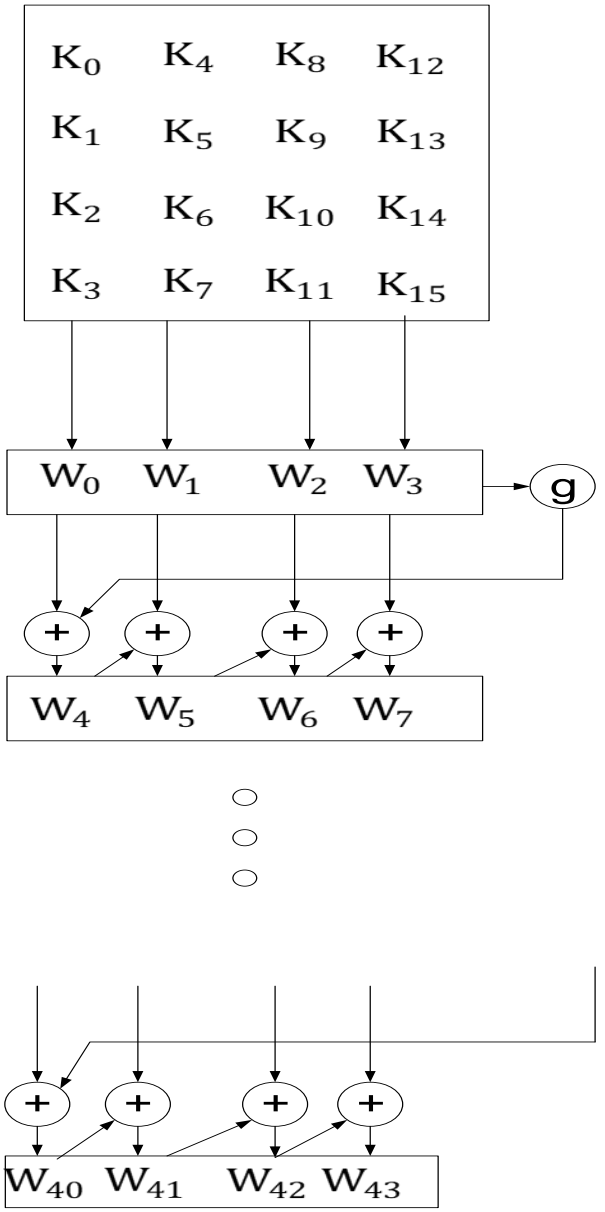
- AES 알고리즘 구조
  - 키 확장(Key expansion)순서
    1. 키를 4바이트씩 나누어 4개의 워드로 만듬
    2. 이전 번의 워드와 4번째전 워드를 XOR하여 새 워드를 생성(4의 배수 번째 경우 이전 워드를 g함수에 통과시킴)
       1. 입력의 워드를 1바이트만큼 왼쪽 순환시프트를 함
       2. Subbytes에서 썼던 치환을 사용
       3. 상수값 R과XOR 하여 결과를 반환
    3. 워드가 44개 생성될때까지 2를 반복

대칭 암호 알고리즘

- AES 알고리즘 구조
  - 키 확장(Key expansion)과정
    3. XOR연산
       - 새로 생성하는 라운드 키 행렬의 첫 번 째 열 생성
         - 새로 생성하는 라운드 키 행렬의 첫 번 째 열과 1,2번을 수행한 결과 값과 Rcon의 라운드 수 번째 열을 XOR연산
       - 라운드 키의 2번 째 열 계산
         - Rcon과 XOR 연산한 값(새로 생성한 행렬의 첫 번 째 열)과 Cipher key의 첫 번 째 열을 XOR 연산
       - 라운드 키 의 3,4번째 계산
         - 새로운 행렬의 열과 기존 Cipher key의 열을 XOR 연산

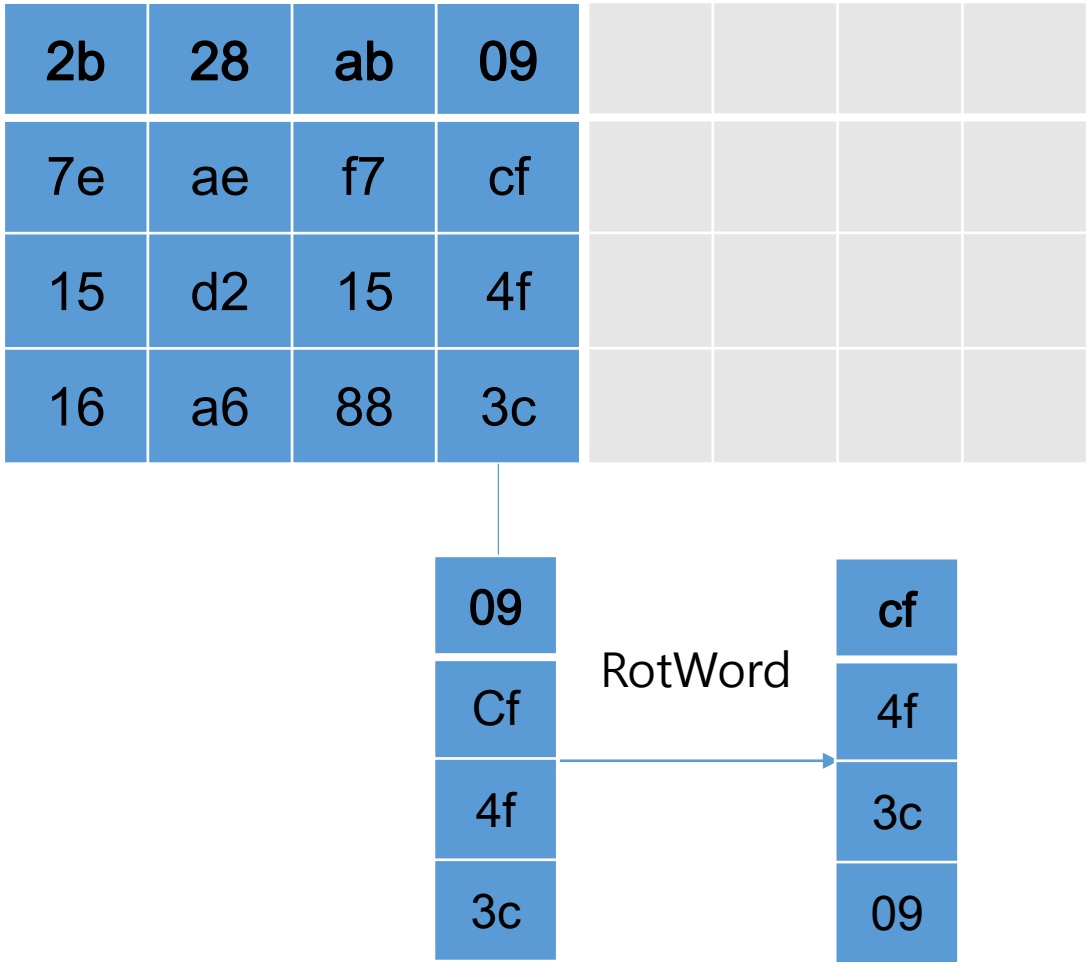
대칭 암호 알고리즘

- AES 알고리즘 구조
  - 키 확장(Key expansion)과정



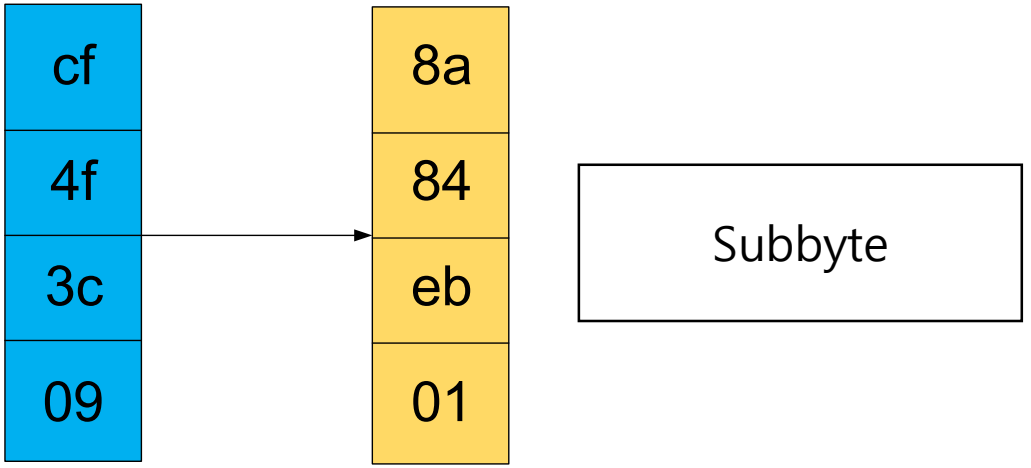
대칭 암호 알고리즘

- AES 알고리즘 구조(그림 수정)
  - 키 확장(Key expansion)과정
    1. RotWord
       - 간단한 순열 변환



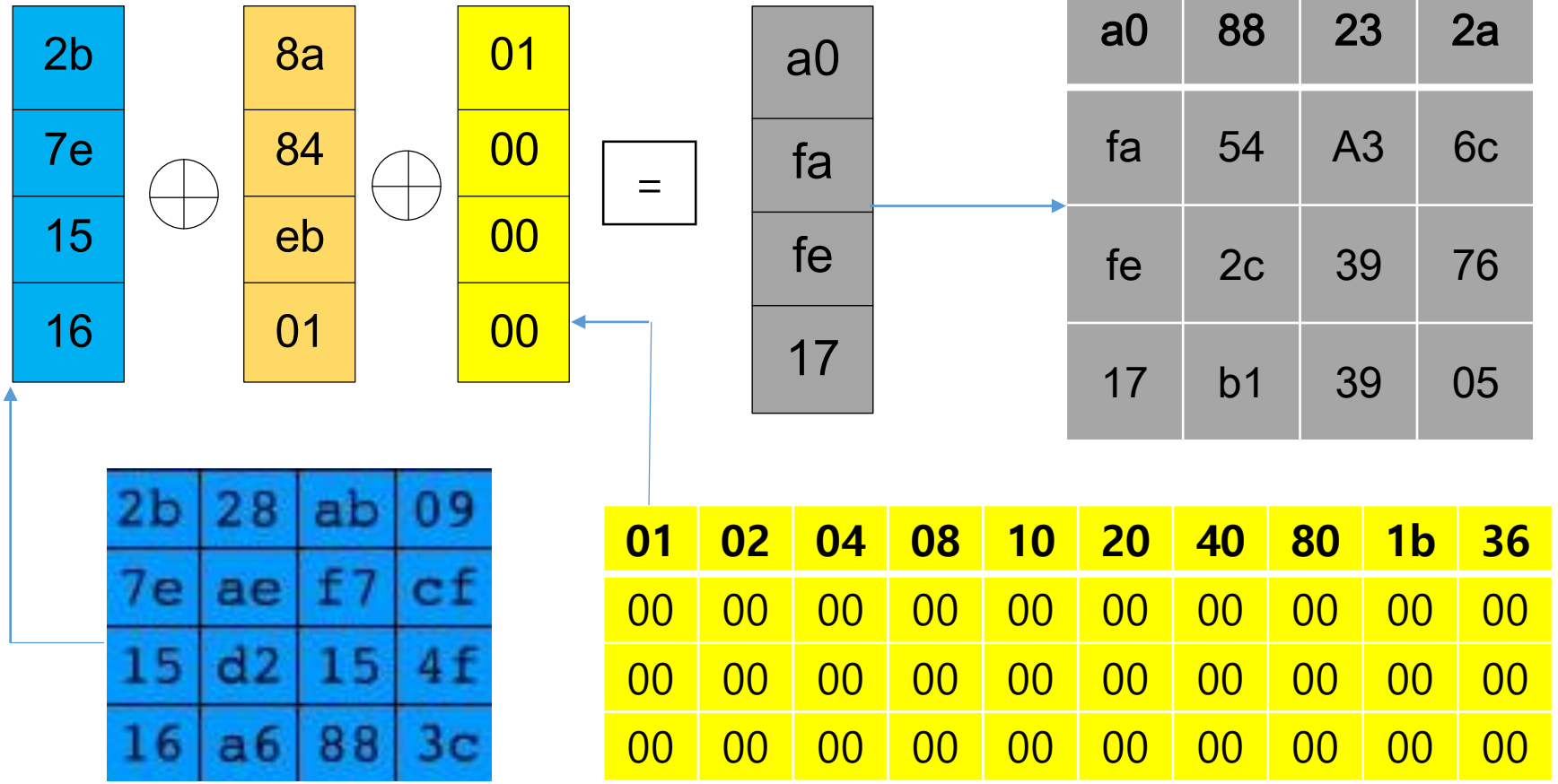
대칭 암호 알고리즘

- AES 알고리즘 구조(그림 수정)
  - 키 확장(Key expansion)과정
    2. Subbyte(바이트 대체)
       - S-Box를 이용해 대체



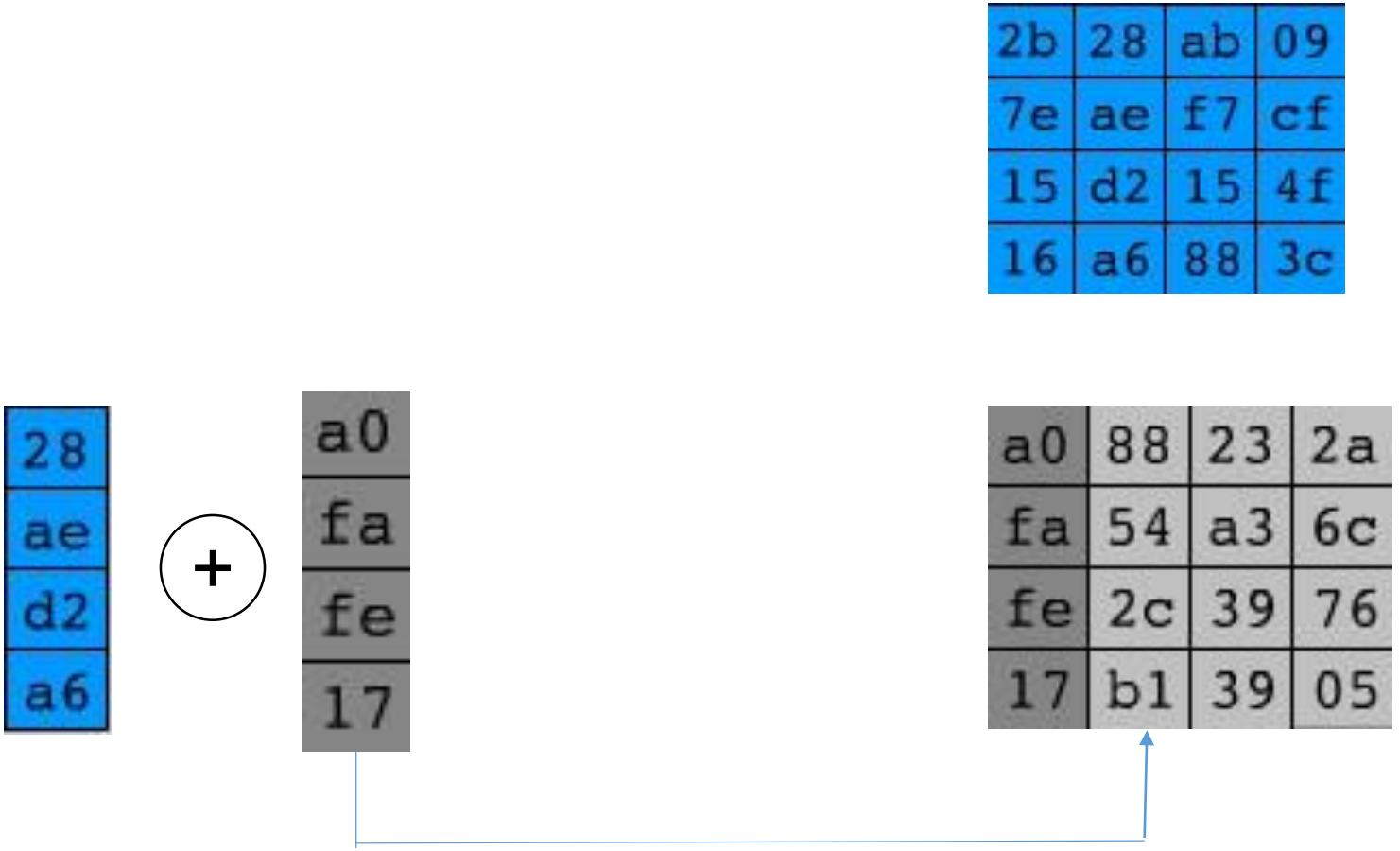
대칭 암호 알고리즘

- AES 알고리즘 구조(그림 수정)
  - 키 확장(Key expansion)과정



대칭 암호 알고리즘

- AES 알고리즘 구조(그림 수정)
  - 키 확장(Key expansion) 예시



랜덤넘버와 의사랜덤넘버

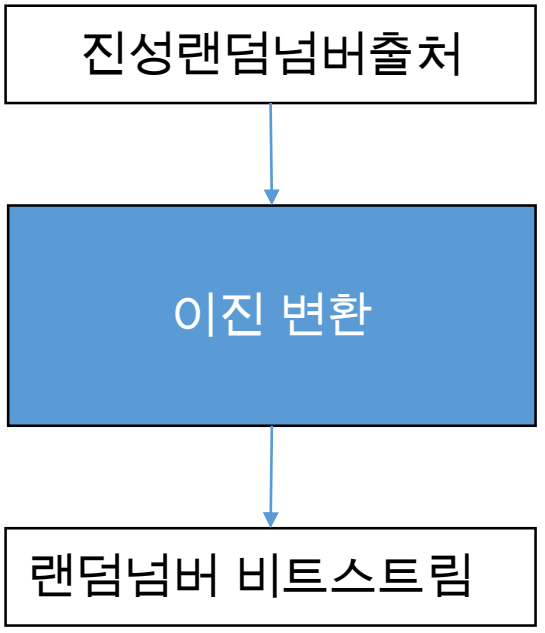
- 랜덤넘버(Random Number)
  - 무작위적으로 추출되어 예측 불가능한 숫자열
  - 암호화에 기반한 다수의 네트워크 보안 알고리즘에서 사용
  - 특징
    - 무작위성(Randomness)
      - 수열이 어느 한쪽으로 치우치지 않고 무작위로 분포되어야 함
      - 균등 분포(Uniform Distribution)
        - 비트열에 나타나는 0과 1비트의 빈도가 균등해야 함
      - 독립성(Independence)
        - 수열에서 추측한 부분 수열이 다른 수열과 연관성이 없어야 함

랜덤넘버와 의사랜덤넘버

- 랜덤넘버(Random Number)
  - 특징
    - 예측 불가능성(Unpredictable)
      - 수열의 일부를 보고 다음에 이어지는 수를 예측할 수 없어야 함
        - e.g., 과거에 알려진 난수열이 공격자에게 알려져도 다음에 출력하는 난수열을 공격자는 알아 맞힐수 없다는 성질

랜덤넘버와 의사랜덤넘버

- TRNG,PRNG와 PRF
  - 진성랜덤넘버 생성기(True Random Number Generator)
    - 엔트로피 소스 사용
      - 암고리즘의 입력으로 엔트로피 소스를 조합하여 사용, 랜덤한 바이너리값을 출력 가능
      - 컴퓨터의 물리적 환경에서 얻을 수 있는 값
        - e.g., 타이밍 패턴, 디스크 전기 작용, 마우스 움직임, 시스템 클록의 순간값

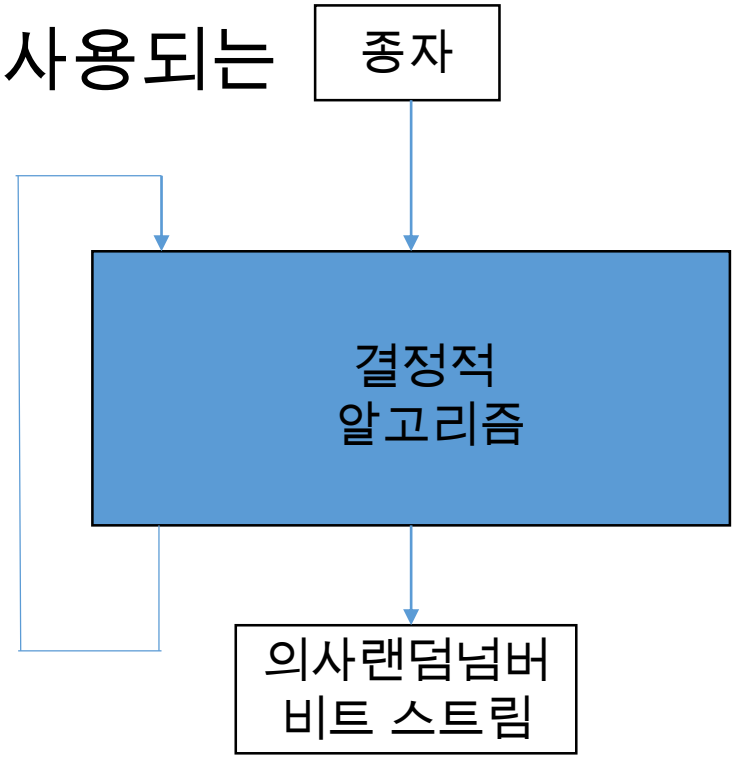


랜덤넘버와 의사랜덤넘버

- 의사랜덤넘버
  - 의사랜덤넘버(Pseudo Random Number)
    - 입력되는 Seed(종자)값에 의해 난수가 결정됨
      - Seed
        - 고정된 값
        - 랜덤한 비트열
        - Seed를 이용해야만 자신만이 사용할 의사난수열 생성 가능
    - 결정적 알고리즘 사용
      - 입력 값(Seed 값)이 같으면 출력 값이 같은 알고리즘
      - 피드백 경로가 있음
        - 출력값을 다시 입력값으로 사용함

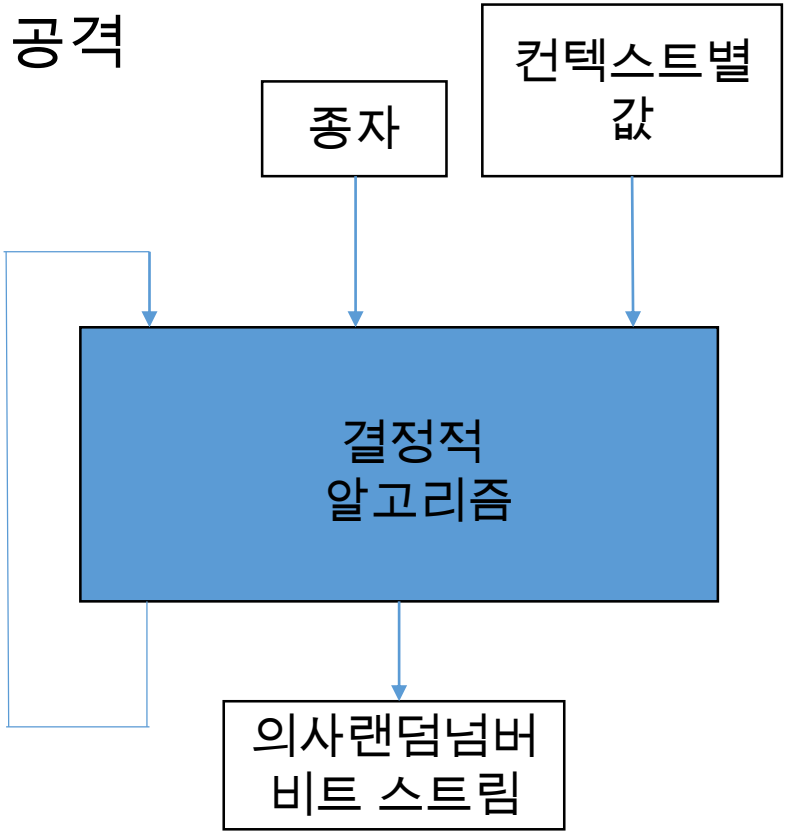
랜덤넘버와 의사랜덤넘버

- TRNG,PRNG와 PRF
  - 의사랜덤넘버 생성기(Pseudo Random Number Generator)
    - 난수를 생성하는 소프트웨어
    - 소프트웨어만으로는 난수를 생성할 수 없어 의사랜덤넘버생성기라 함
    - 무한 비트열 생성하기 위해 사용되는 알고리즘



랜덤넘버와 의사랜덤넘버

- TRNG,PRNG와 PRF
  - 의사랜덤넘버 생성기(Pseudo Random Number Function)
    - 고정된 길이의 의사 랜덤 비트열을 생성하기 위해 사용되는 함수
      - SALT값:암호화 방법을 풀기 위해서 공격할때 딕셔러리 만들어서 다른 암호화 예제와 비교 하는걸 막기 위한 방법 키값으로 알아도됨
    - PRNG와 생성되는 비트열만 다름



감사합니다!