2020/01/10, 2020보안 기초 세미나

Network Security Essentials

- Chapter 2 대칭 암호와 메시지 기밀성(1) -

박 재 형(jaehyoung@pel.smuc.ac.kr)

상명대학교 프로토콜공학연구실

목 차

- 대칭 암호 원리
- 대칭 암호 알고리즘
- 랜덤 넘버와 의사 랜덤 넘버

대칭 암호 원리

- 대칭 암호
  - 메시지를 암호화와 복호화에 같은 암호 키를 사용하는 방식
- 대칭 암호 구조 (1/2)
  - 평문 (Plaintext)
    - 누구나 읽을 수 있는 문서나 데이터로서 알고리즘의 입력으로 이용됨
  - 암호화 (encryption)
    - 평문을 알아 볼 수 없도록 암호문으로 변환 하는 방식
  - 암호문 (Ciphertext)
    - 평문이 암호 알고리즘에 의해 암호화된 메시지
  - 비밀 키 (Secret key)
    - 평문을 암/복호화 시키는 핵심 가변 정보 값

대칭 암호 원리

- 대칭 암호
  - 메시지를 암호화와 복호화에 같은 암호 키를 사용하는 방식
- 대칭 암호 구조 (2/2)
  - 암호 알고리즘 (Encryption algorithm)
    - 평문을 암호화하여 암호 문으로 변화시키는 알고리즘
  - 복호 알고리즘 (Decryption algorithm)
    - 암호 알고리즘을 역으로 수행하는 것과 같음
      - 암호문에 적용된 비밀 키를 사용하여 평문으로 복구하는 것

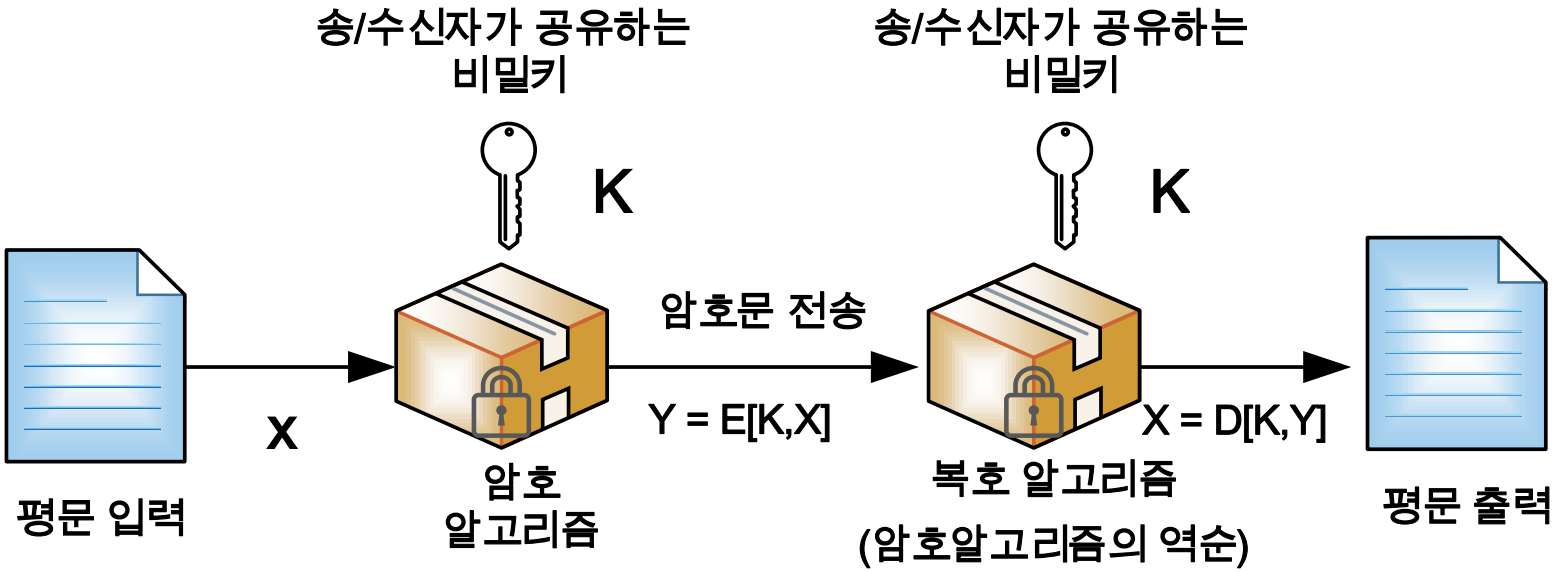
대칭 암호 원리

- 대칭키 암호 보안의 필수 사항
  - 해독하기 어려운 강한 알고리즘이 있어야함
    - 공격자가 절대로 암호문을 해독하거나 알아 낼 수 없어야 함
  - 송신자와 수신자는 공유하는 비밀 키를 안전한 방법으로 획득하고 보관해야 한다
    - e.g., Kerberos같은 중앙 기관이 키 교환을 도와줌
      - Kerberos란 사용자의 인증을 위한 프로토콜

대칭 암호 원리

- 대칭키 암호 보안의 필수 사항
  - 대칭 암호의 보안은 키의 비밀성에 의해 지켜 짐
  - 암호 알고리즘은 공개 되도 키는 무조건 비밀로 해야 함
    - 공격자가 암호-복호 알고리즘과 암호문을 알고있어도 그 암호문을 해독 할 수 없어야 하기 때문임

- 대칭 암호 모델



대칭 암호 원리

- 암호 시스템의 3가지 (1/2)
  - 평문을 암호문으로 전환하는 연산 유형
    - 대체 (Substitution)
      - 평문에 각 요소(비트, 문자, 블록)를 다른 요소 로 바꾸어 암호화
        - e.g.,(가, 나, 다,라) ⟶ (아, 자, 타,파)
    - 전치 (Transposition)
      - 요소의 순서를 재조정 하는 것
        - e.g.,(가, 나, 다, 라) ⟶ (라, 나, 가, 다)
  - 사용되는 키
    - 송신자와 수신자가 같은 키를 사용한다면 대칭암호라고 부르며 다른 키를 사용하면 공개키 암호라고 부름

대칭 암호 원리

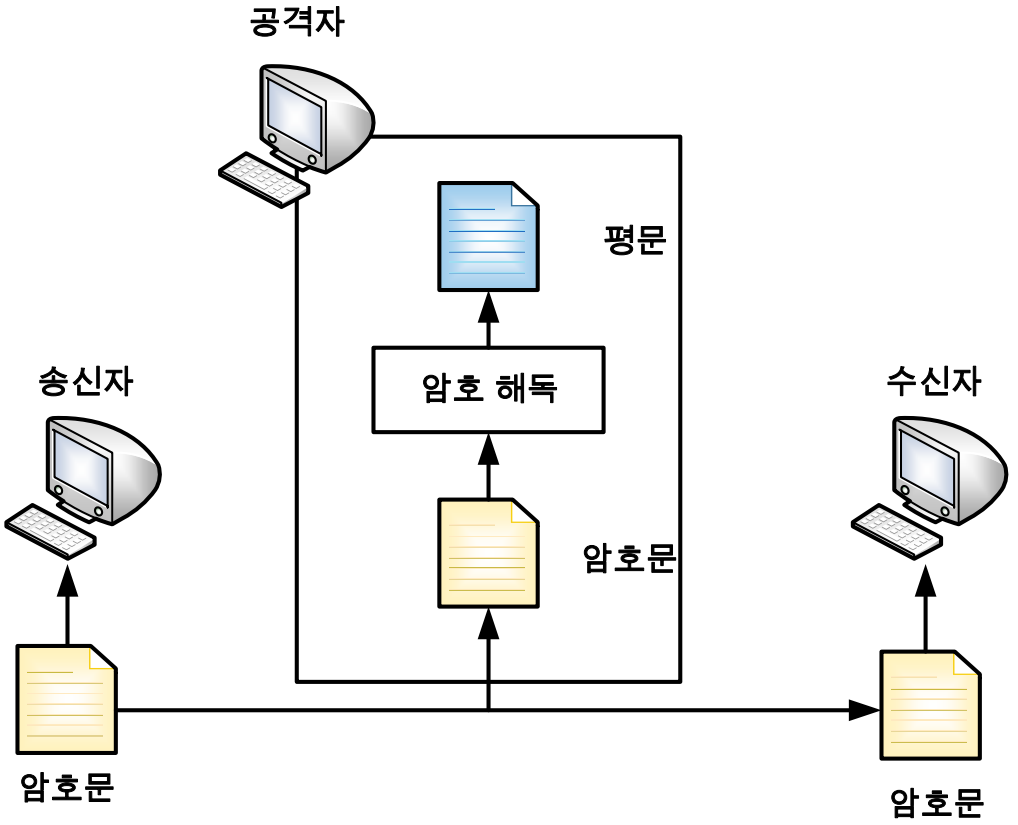
- 암호 시스템의 3가지 (2/2)
  - 평문이 처리되는 방법
    - 블록 암호(Block cipher)
      - 블록은 특정 비트 수의 집합을 뜻함
      - 데이터를 정해진 블록 단위로 암호화 하는 것
        - e.g.,메시지 전송
    - 스트림 암호(Stream cipher)
      - 유사 난수를 1비트 단위로 연속적으로 생성하여 암호화 하는 것
        - e.g.,오디오/비디오 스트리밍

대칭 암호 원리

- 암호 해독
  - 의미
    - 평문이나 키를 찾으려는 시도
  - 특징
    - 해독 전략은 암호 구조와 해독가의 정보(주어진 상황, 암호 지식, 장비 등) 에 따라 달라짐

대칭 암호 원리

- 암호화된 메시지 공격 유형(1/5)
  - 암호문 단독 공격 (Ciphertext-only-attack)
    - 단지 암호문만 가지고 평문이나 키를 찾아내는 공격 방법
    - 공격자가 소지한 정보가 부족하여 공격자가 불리 함

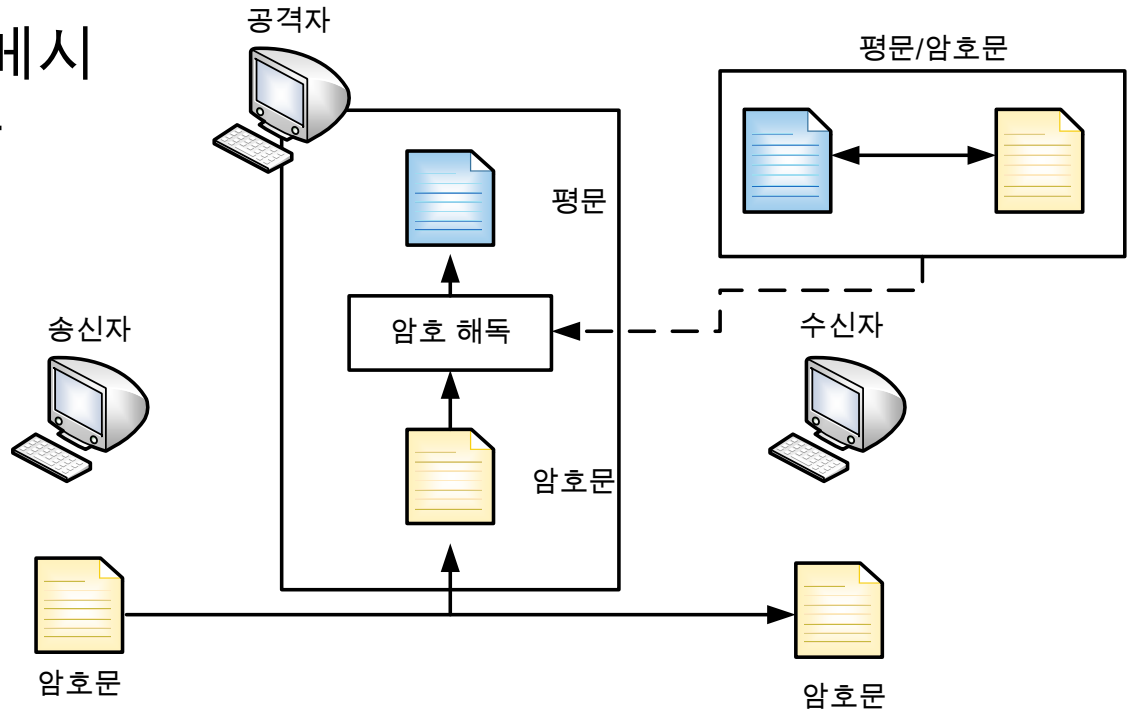


대칭 암호 원리

- 암호화된 메시지 공격 유형(2/5)
  - 암호문 단독 공격의 유형
    - 전수 조사 공격 (Brute Force attack)
      - 암호문을 해독 할 수 있는 모든 경우를 시도해보는 공격
    - 빈도 분석 공격 (frequency analysis attack)
      - 암호문에서 통계적으로 많이 사용되는 문자 또는 문자열의 사용빈도 분석하여 정보를 얻어내어 암호문을 해독하는 공격
    - 패턴 공격 (Pattern attack)
      - 암호문에 존재하는 패턴을 이용하여 평문을 유추하는 공격
        - e.g.,포스트스크립트(텍스트 편집기)로 부호화된 파일
          - 항상 특정한 패턴을 가지고 시작됨

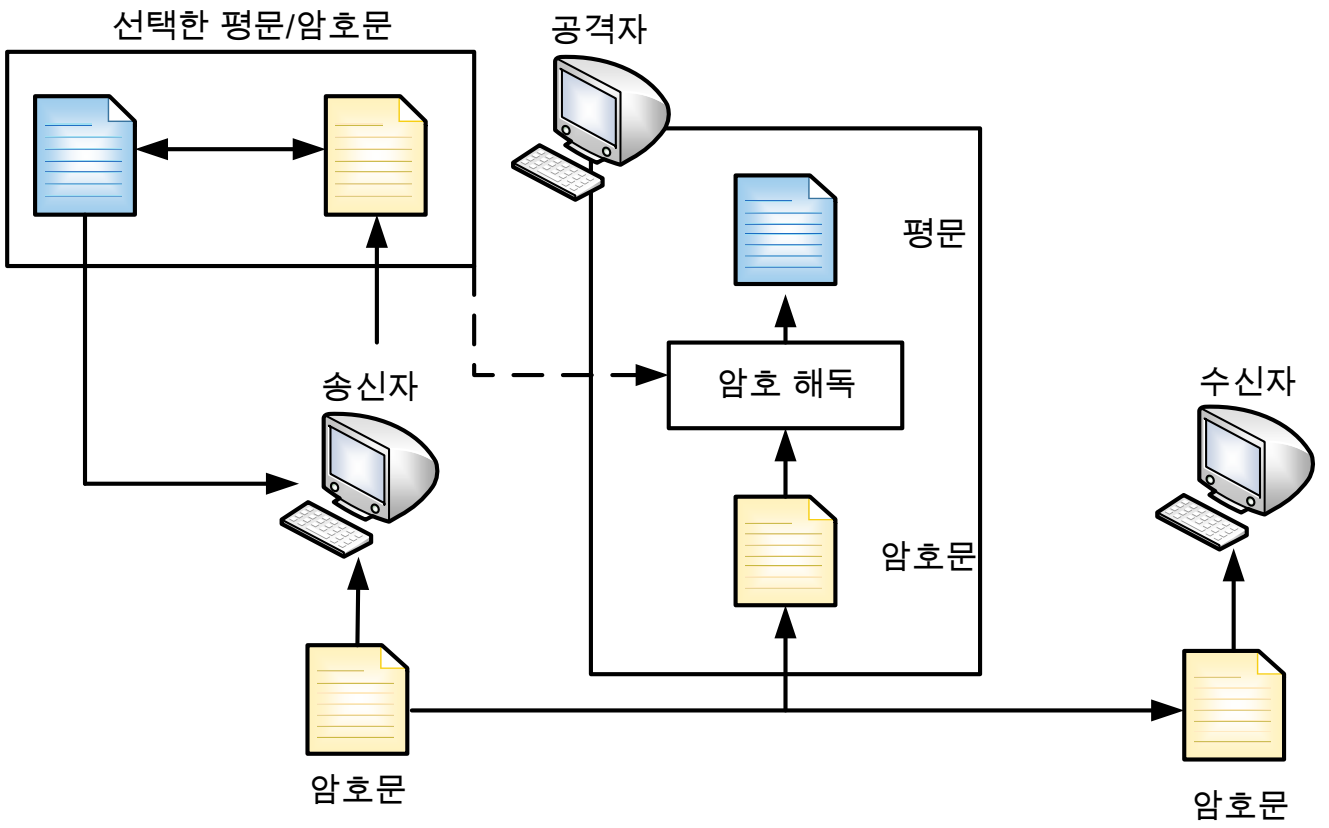
대칭 암호 원리

- 암호화된 메시지 공격 유형(3/5)
  - 알려진 평문 공격 (Known plaintext attack)
    - 평문과 암호문 조합 알려져 있어 키와 전체 평문을 추정하여 해독하는 공격 방법
      - 암호문 단독 공격과 마찬가지로 전수조사, 통계적 분석, 패턴 공격을 사용 할 수 있다
      - e.g., 전자 금융 거래 메시지는 표준화된 헤더나 제목이 붙어있음



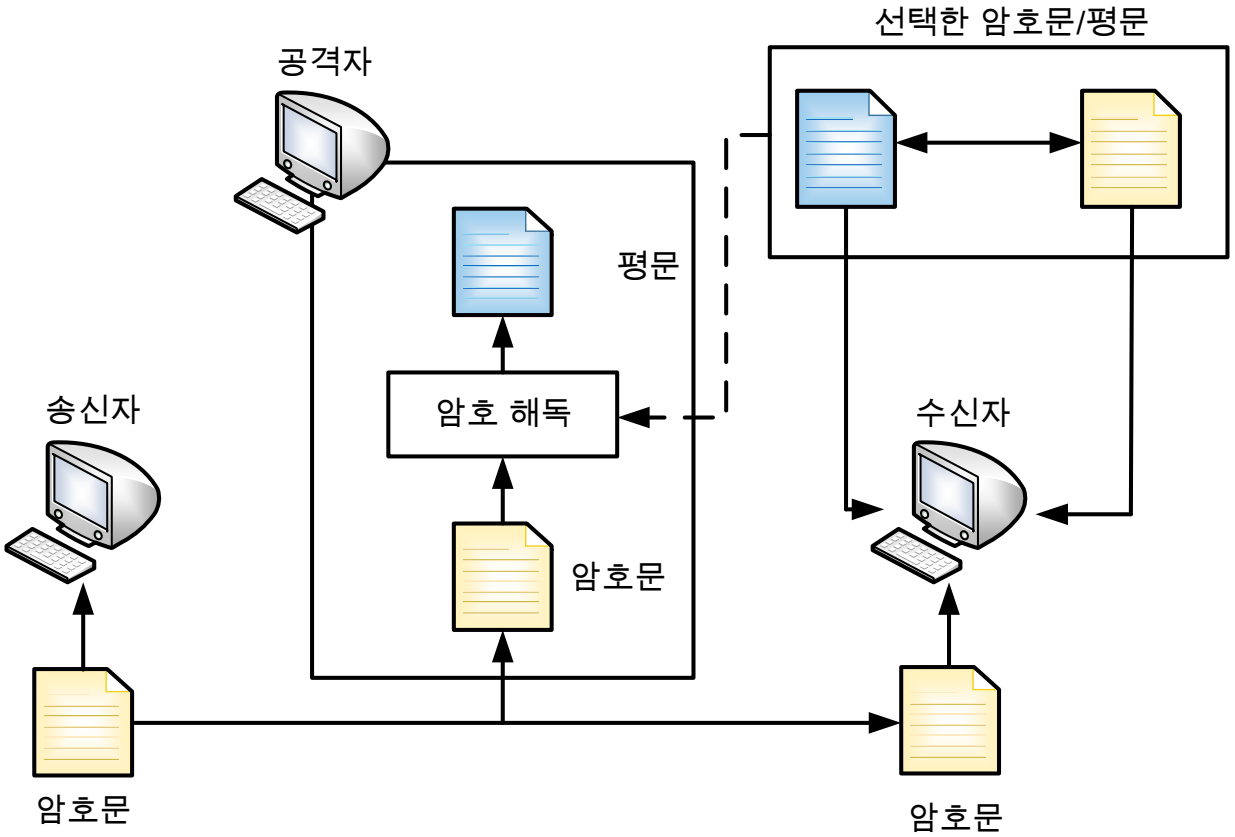
대칭 암호 원리

- 암호화된 메시지 공격 유형(4/5)
  - 선택 평문 공격(Chosen-plaintext attack)
    - 공격자가 송신자의 시스템 근원지에 접속하여 평문을 입력하고 암호화 시켜보는 공격 방법



대칭 암호 원리

- 암호화된 메시지 공격 유형(5/5)
  - 선택 암호문 공격(Chosen-ciphertext attack)
    - 공격자가 송신자의 시스템 근원지에 접속하여 암호문을 획득하여 획득한 암호문을 제외한 다른 암호문을 해독하는 공격 방법



대칭 암호 원리

- 암호 구조가 계산적으로 안전한 구조
  - 암호문을 깨는 데 드는 비용이 암호화된 정보의 가치보다 더 큼
  - 암호문을 깨는데 걸리는 시간이 해당 정보의 수명보다 더 김

- 전수 공격시 키 탐색에 요구되는 평균 시간

키 크기(비트)	키의 종류 수	μs당 한 번의 암호화를 할 때 소요되는 시간	μs당 106번의 암호화를 할 때 소요되는 시간
32	2^{32} = 4.3x10^{9}	$2^{31}\mu s$ = 35.8분	2.15μs
56	2^{56} = 7.2x10^{16}	$2^{55}\mu$s = 1142년	10.01시간
128	2^{128} = 3.4x10^{38}	$2^{127}\mu$s = 5.4x10^{24}년	5.4x10^{18}년
168	2^{168} = 3.7x10^{50}	$2^{167}\mu s$ = 5.9x10^{30}년	5.9x10^{30}년
26개 문자(치환)	26! = 4x10^{26}	2x$10^{26}\mu s$ = 6.4x10^{12}년	6.4x10^{6}년

대칭 암호 원리

- Feistel 암호 구조
  - 1973년 IBM(International Business Machines)의 Horst Feistel이 최초로 소개한 암호 구조
  - 특징
    - 대체와 변환을 번갈아 사용하여 메시지를 암/복호화
    - 대부분의 블록 암호 알고리즘의 구조는 Feistel 암호 구조를 따라 만들어 짐
      - e.g.,DES, 3DES
    - 여러 개의 라운드로 이루어짐
      - 기본적으로 16라운드 사용, 사용자가 라운드 수 조정 가능
    - 블록 크기와 길이가 길수록 안전성이 높음
      - 암/복호화 속도가 떨어짐

대칭 암호 원리

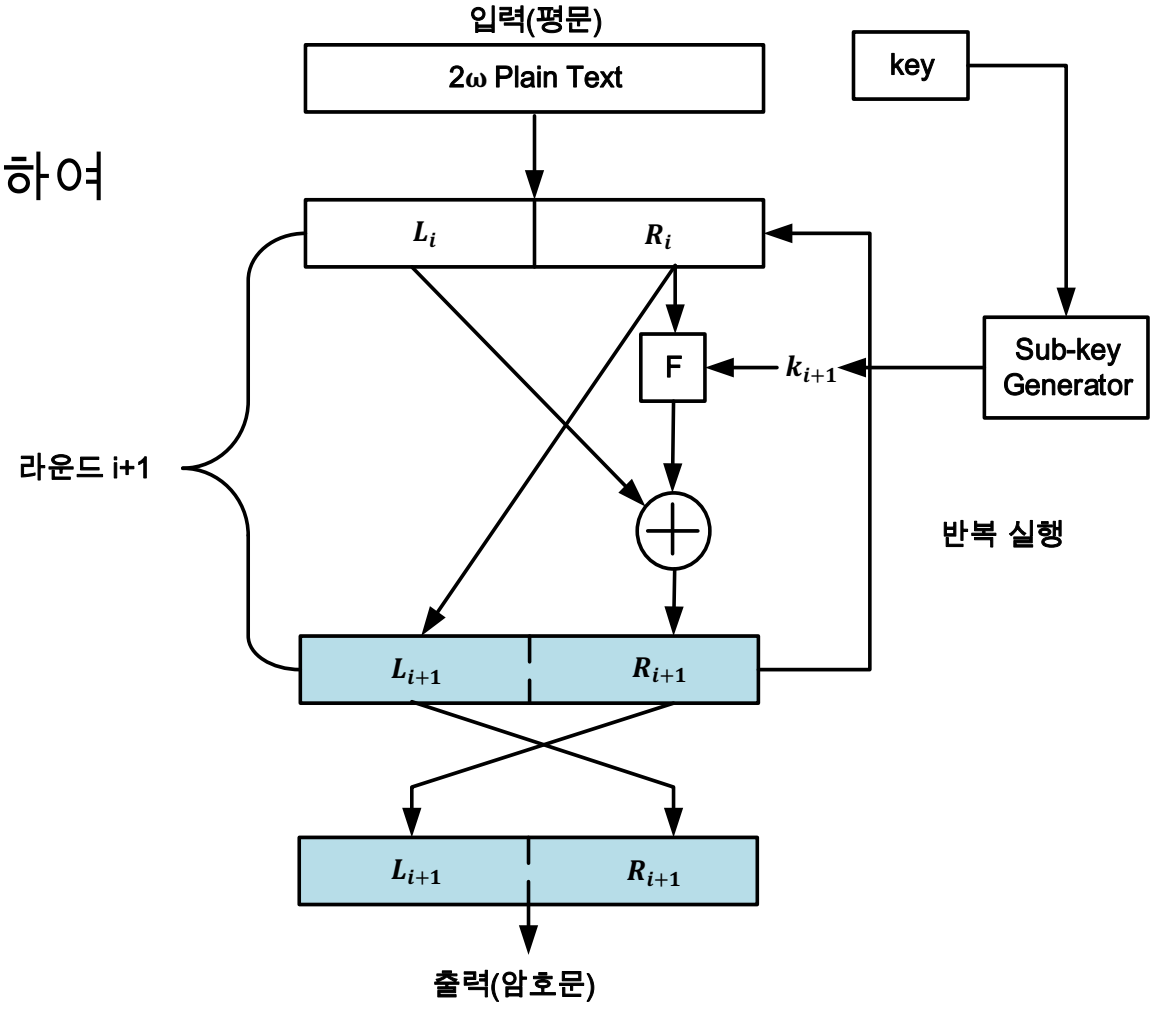
- Feistel 암호 구조
  - 매개 변수와 설계 특성

	블록 크기 (Block size)	키 크기 (Key size)	라운드 수 (Number of rounds)	서브 키 생성 알고리즘 (Subkey generation algorithm)	라운드 함수 (Round function)
의미	문자열의 길이	키에 사용되는 bits 수	수행 과정의 수	서브 키를 생성하는 알고리즘	수행 과정에 반복되는 함수
설계 특성	1.길이가 길면 더 강한 보안을 의미 2.암/복호화 시간이 오래 걸림 3.64bits가 합리적인 크기	1.길이가 길면 더 강한 보안을 의미 2.암/복호화 시간이 오래 걸림 3.128bits가 합리적인 크기	1.라운드 수를 증가 시켜 여러 번 수행하면 보안을 강화 시킬 수 있음 2.전형적인 라운드 수는 16임	이 알고리즘이 복잡할수록 암호 해독이 어려워 짐	이 알고리즘이 복잡할수록 암호 해독이 어려워 짐

대칭 암호 원리

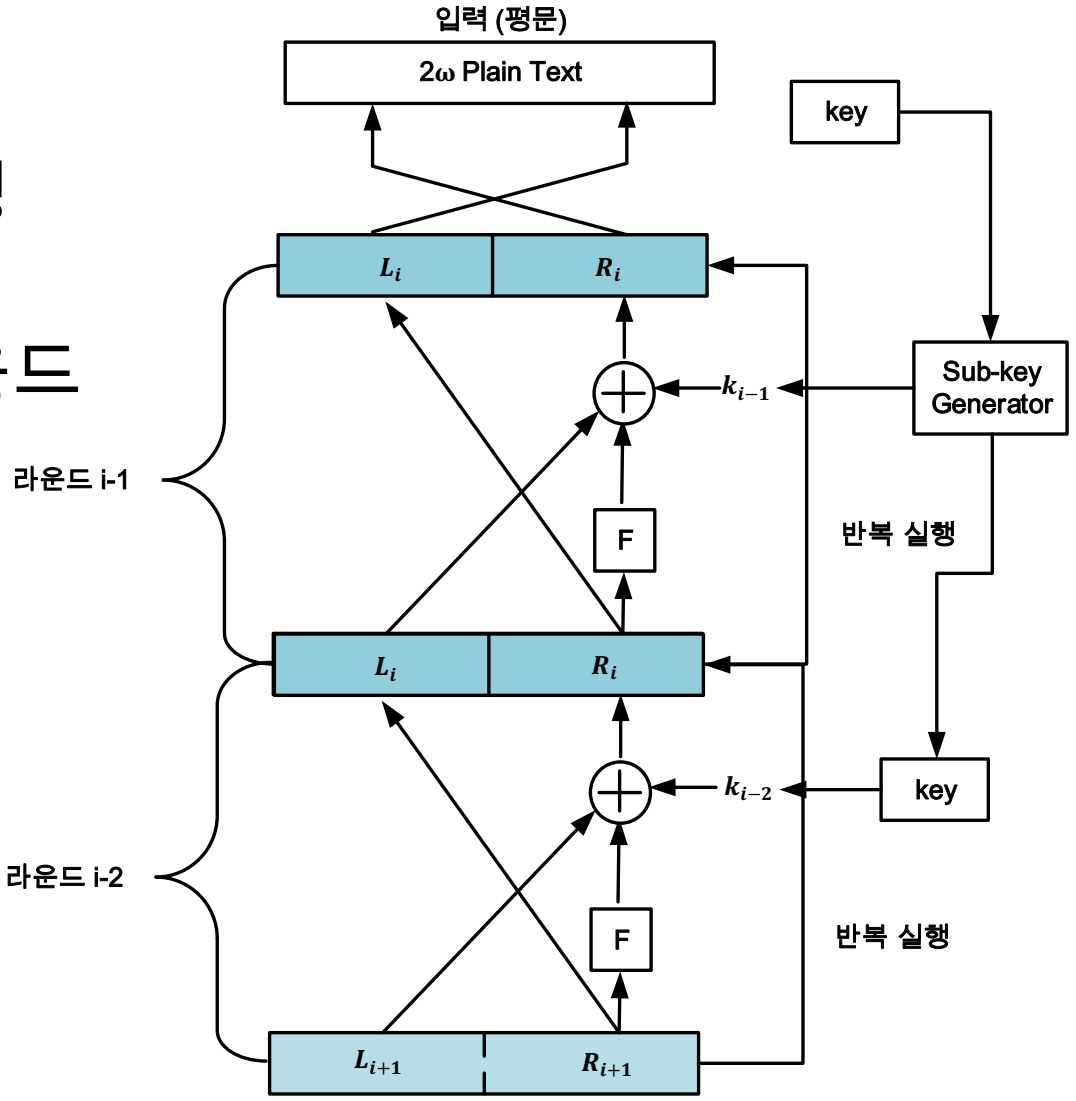
- Feistel 암호 구조

  - Feistel 암호화 과정
    - 하나의 평문 블록을 입력2ω
      - 성능과 속도 효율 을 고려하여 64bits권장
    - 평문 블록을 반 씩 나눔
    - $F(R_i, K_{i+1}) \oplus L_i = R_{i+1}$
      - 치환과 대체를 수행하는 함수
    - 위 과정이 1회에 1라운드이며 16회 반복 실행
    - 마지막 라운드가 끝나면 R_{16}의 위치를 교환하여 암호문 출력



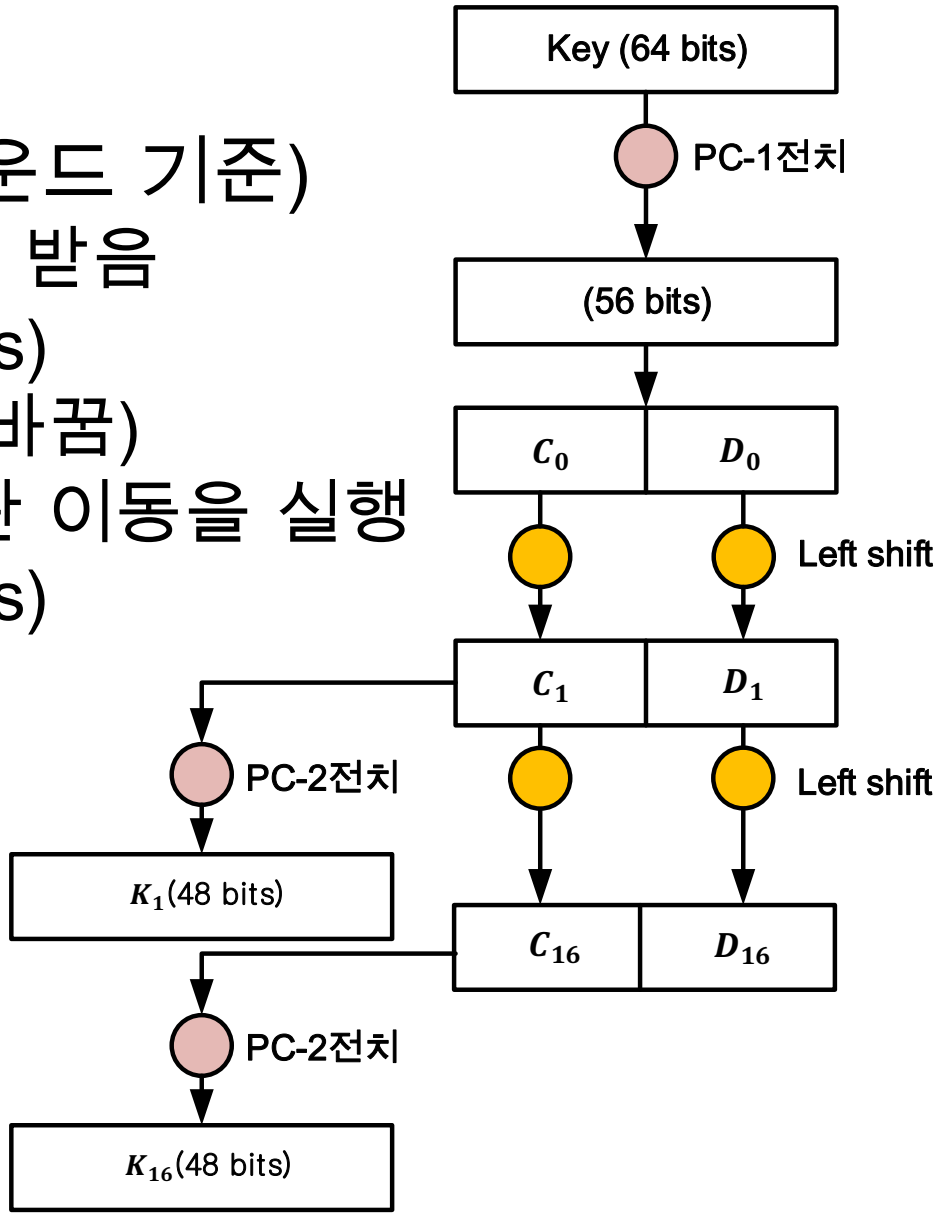
대칭 암호 원리

- Feistel 암호 구조
  - Feistel 복호화 과정
    - 암호화의 역순으로 진행
    - 64 bits평문 기준 16라운드 진행
    - 마지막 라운드 진행 후 L_i와R_i의 자리 교환



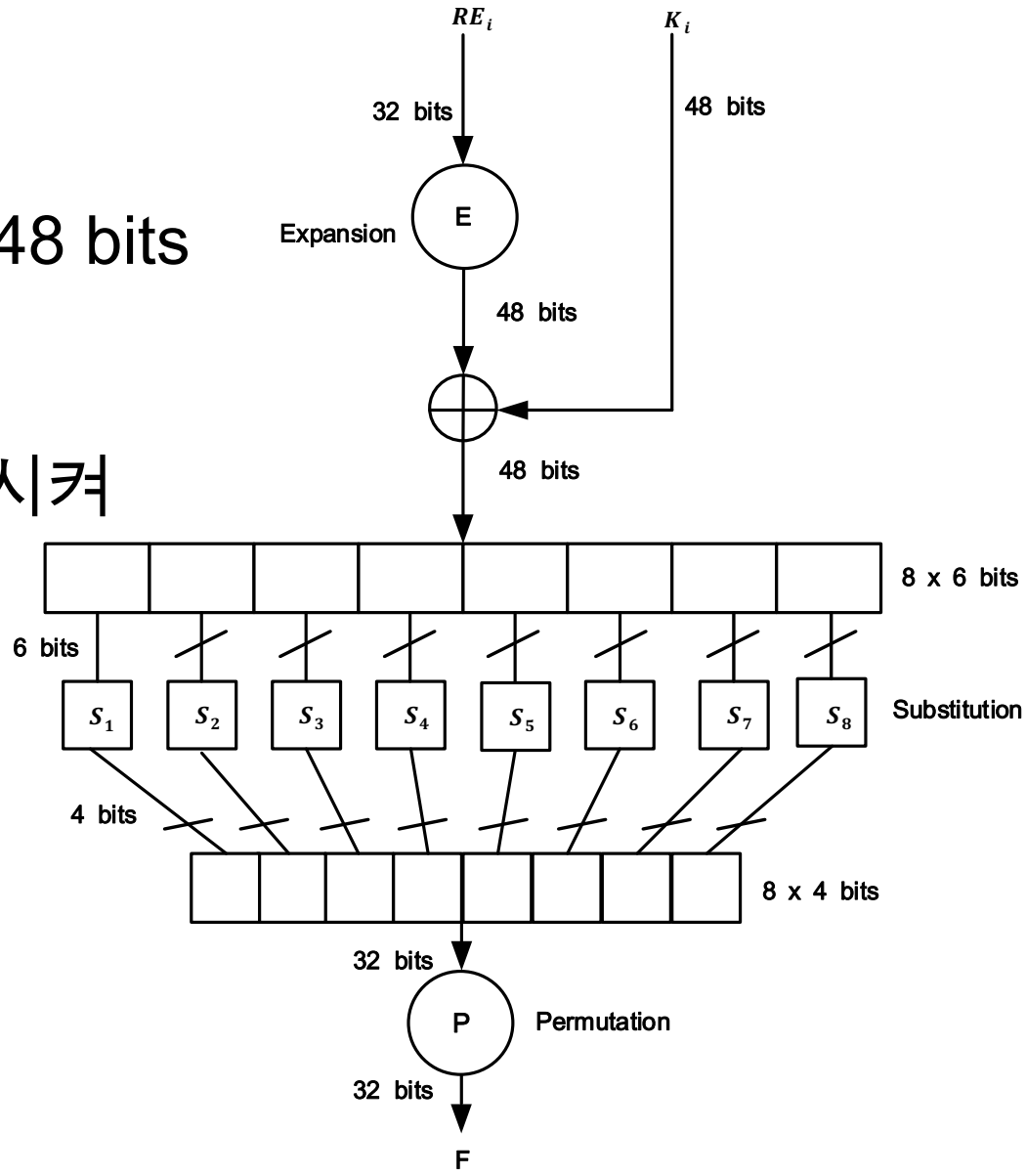
대칭 암호 원리

- Feistel 암호 구조
  - 서브 키 (Sub key) (16라운드 기준)
    - 64 bits크기의 키를 입력 받음
    - 패리티 비트 제거(56 bits)
    - 전치 (데이터의 순서를 바꿈)
    - 반으로 나누어 좌측 순환 이동을 실행
    - 패리티 비트 제거(48 bits)
    - 전치
    - 키 생성
    - 위 과정을 16회 반복
    - 서브 키는 모두 다르다



대칭 암호 원리

- Feistel 암호 구조
  - F함수 (라운드 함수)
    - 32 bits 값을 확장(E)하여 48 bits 로만들어 XOR연산을 함
    - 48 bits를 8개로 쪼갬
    - 6 bits 값이 S_1~S_8을 통과 시켜 4 bits로 변환
    - 최종 32 bits를 P연산을 하여 F함수 값이 나옴



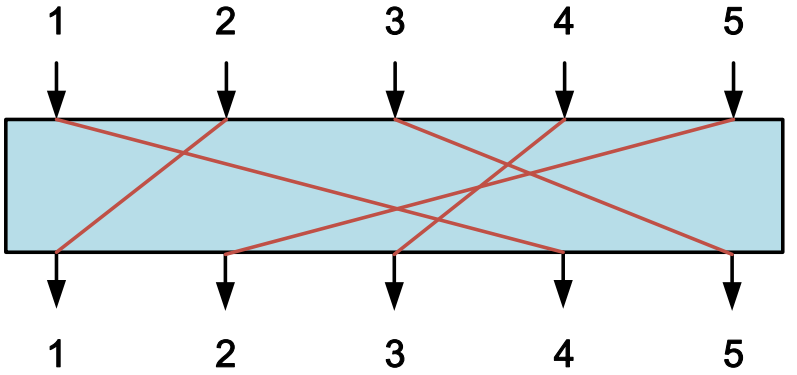
대칭 암호 원리

- Feistel 암호 구조
  - S-box (Substitution-Box)
    - 6 bits의 입력을 4 bits의 출력으로 축소시켜 변환하는 함수
      - S-box 사용
    - 역 방향으로 복원이 어려움
  - S-Box table
    - e.g., 110001(외부 bits = (1,1), 중간 4 bits = (1000)) = 0110

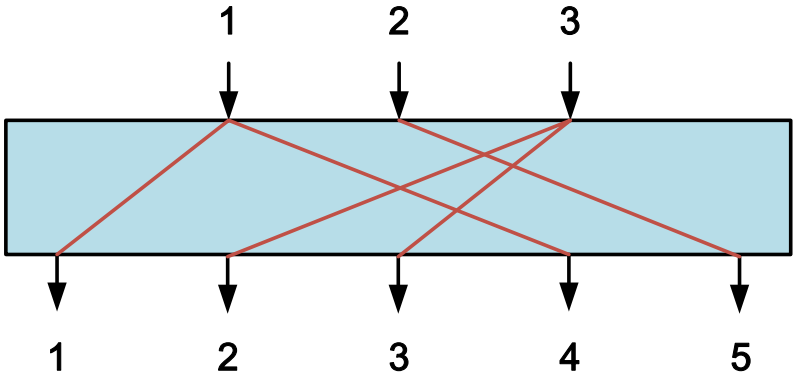
S		중간 4 bits 입력															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
외부 bits	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	1010	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

대칭 암호 원리

- Feistel 암호 구조
  - P-box (permutation-box)
    - 단순 (Straight) P-box
      - n bits를 입력 받아 변화된 m bits를 출력 함



    - 확장 (Expansion) P-box
      - n bits를 입력 받아 변화된 m bits를 출력 함(n < m)
      - Bits를 치환하고 양을 늘릴 때 사용함



대칭 암호 원리

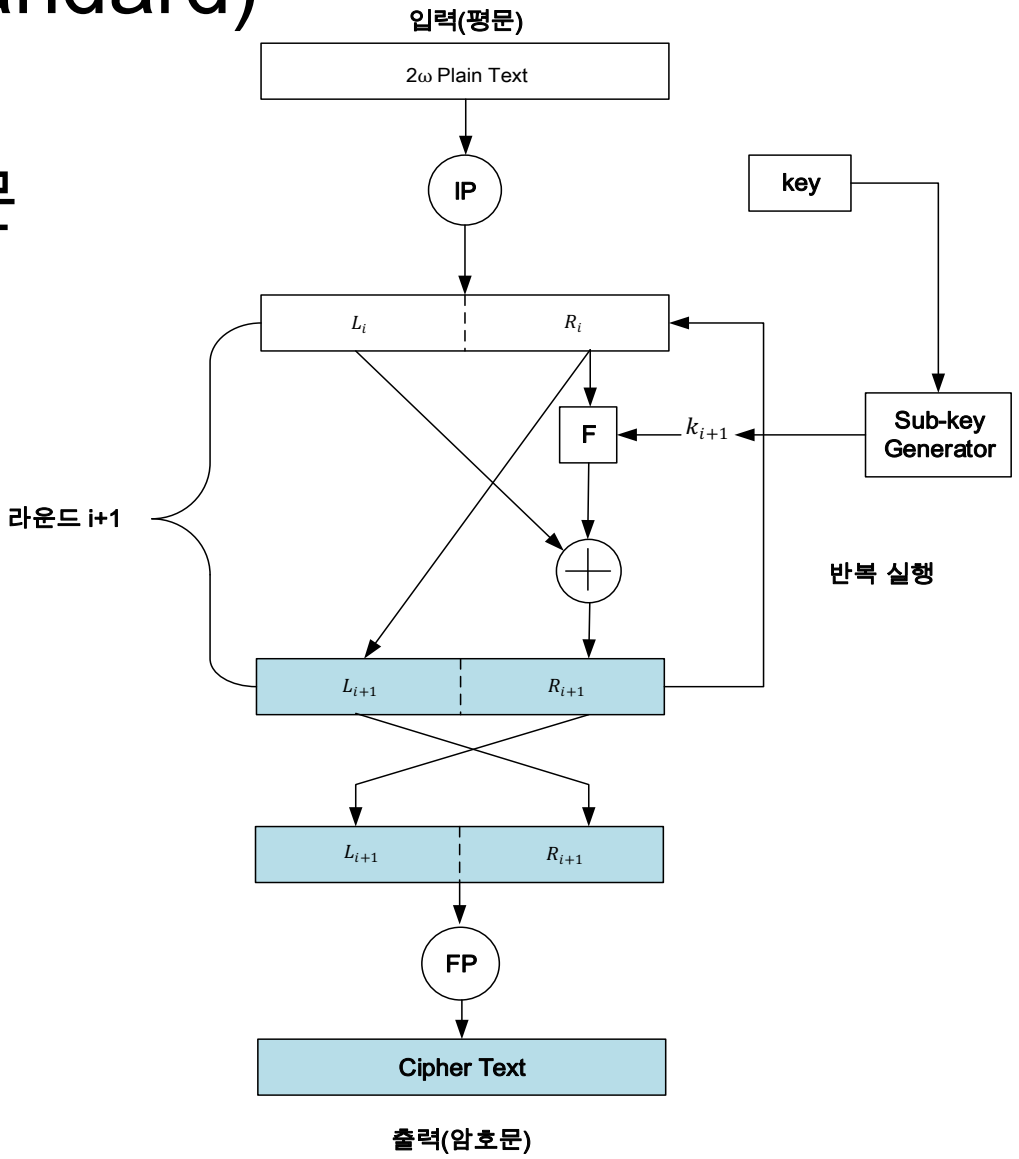
- Feistel 암호 설계에 고려 해야 할 점
  - 알고리즘 실행 속도를 고려 해야 함
    - 빠른 소프트웨어 암호/복호
    - 너무 느리면 가용성이 떨어짐
  - 알고리즘 구조를 용이하게 만들어야 함
    - 알고리즘 구조를 단순히 만들어 취약점을 찾기 쉬워져 더욱 강한 알고리즘을 만들 수 있음

대칭 암호 알고리즘

- DES (Data Encryption Standard)
  - 1972년 미국 국가기술표준원 (NIST: National Institute of Standards and Technology)이 개발한 미국 정부 규모의 표준적인 암호 알고리즘
  - 특징
    - 평문의 길이 64bits
    - 키의 길이 56bits
    - 라운드 횟수 16회
    - Feistel 암호 알고리즘 구조를 취함
    - 각 회전 마다 서브 키를 사용 함
    - 길이가 56bits인 키로부터 길이가 48bits인 서로 다른 16개의 서브 키가 생성 된다

대칭 암호 알고리즘

- DES (Data Encryption Standard)
  - 암호화 과정
    - 입력으로 들어온 64bits평문 블록은 초기치환 IP과정을 거친 후 L_i와 R_i으로 32bits 씩 나뉨
    - F(R_i, K_{i+1})$\oplus L_i$=R_{i+1}
    - 마지막 라운드가 끝나면 L_{16}과R_{16}의 위치 교환
    - 위치 교환 후 평문 블록은 최종 치환FP를 거쳐 64bits 암호문으로 출력됨

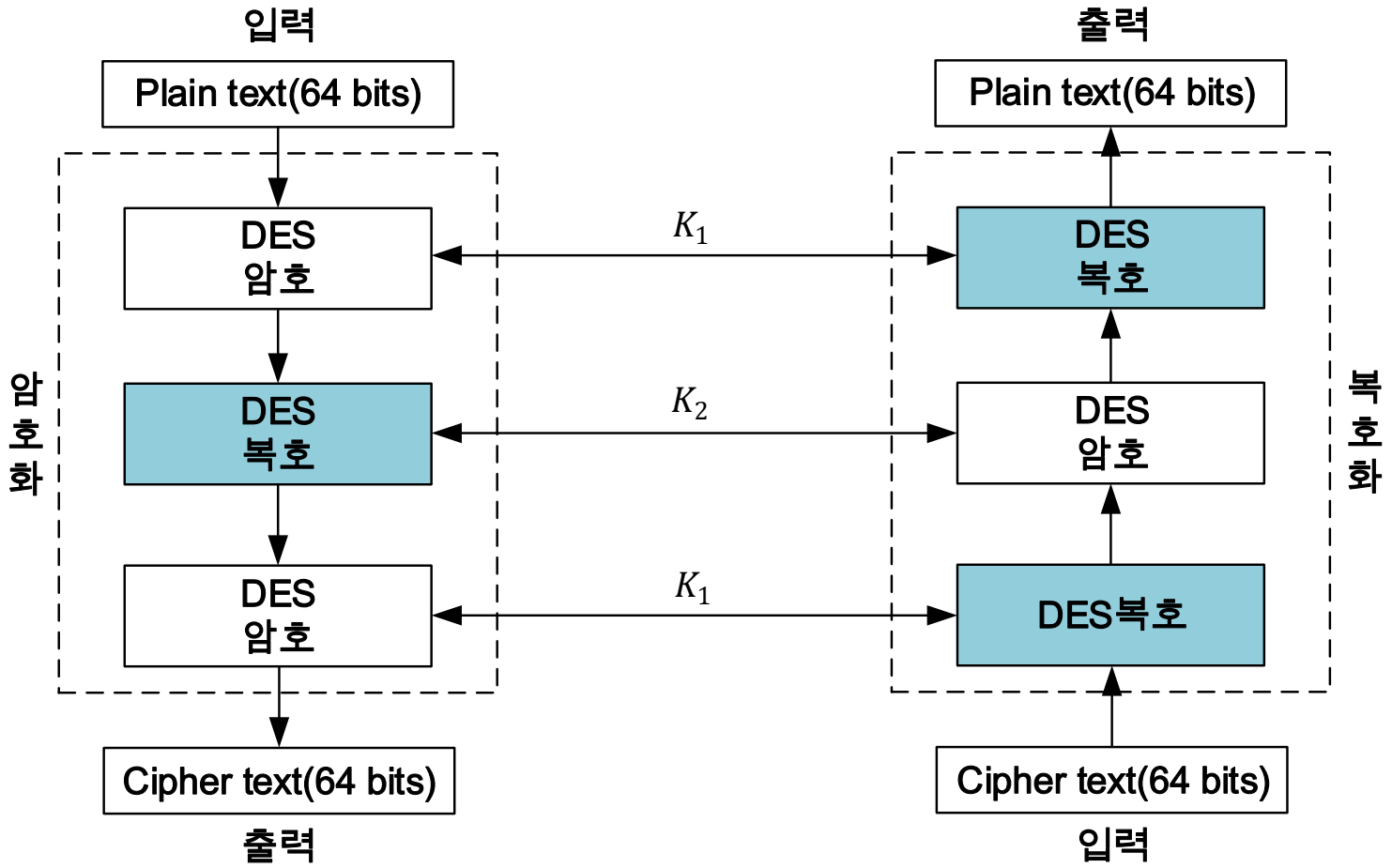


대칭 암호 알고리즘

- 3DES (Triple Data Encryption Standard)
  - DES의 안전성을 향상 시키기 위해 암호화와 복호화에 대하여 DES를 세번 사용한 알고리즘
  - 특징
    - 키의 길이가 168bits가 되기 때문에 전수 공격 차단
    - 라운드 횟수가 3배가되어 속도가 느림
    - 2개의 키를 갖는 3DES
      - 서브키 2개를 사용하여 DES의 암/복호화에 3회 사용
    - 3개의 키를 갖는 3DES
      - 서브키 3개를 사용하여 DES의 암/복호화에 3화 사용

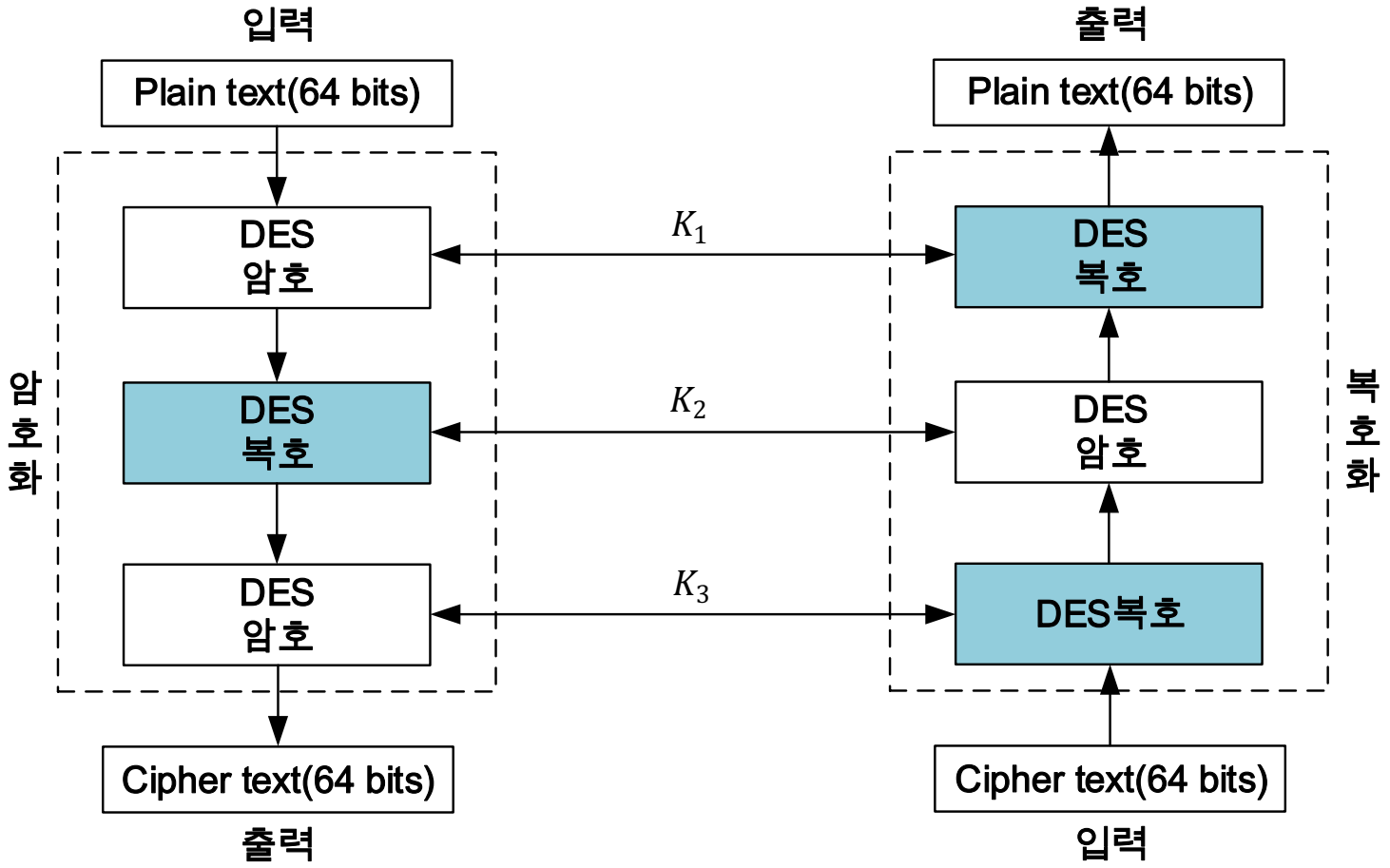
대칭 암호 알고리즘

- 3DES (Triple Data Encryption Standard)
  - 2개의 키를 갖는 DES



대칭 암호 알고리즘

- 3DES (Triple Data Encryption Standard)
  - 3개의 키를 갖는 DES

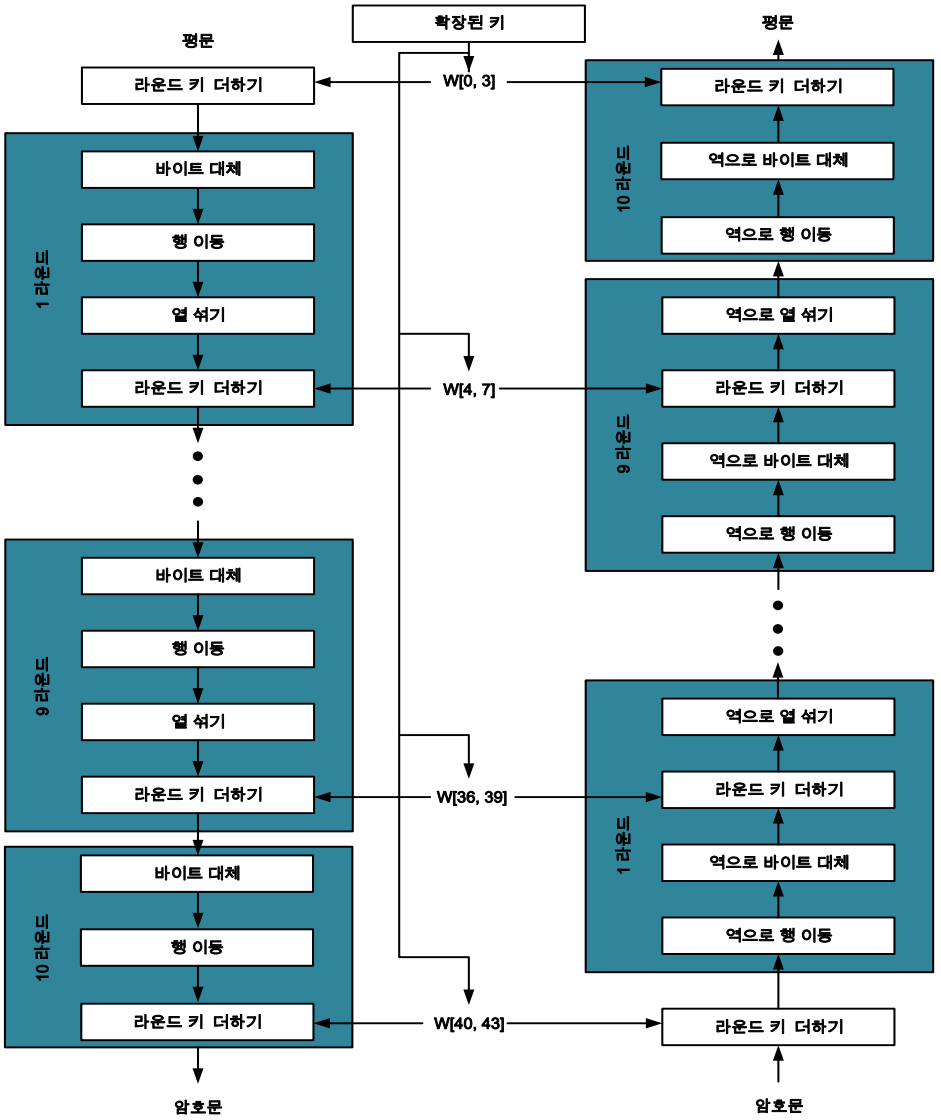


대칭 암호 원리

- AES (Advanced Encryption Standard)
  - 2001년 미국 국립기술표준원 (NIST)에서 공표한 대칭 키 암호 알고리즘
  - 특징
    - 128bits 블록의 평문을 사용 함
    - 키는 128, 192, 256bits 선택해서 사용 가능 함
    - 라운드 수는 키의 길이에 따라 다름
      - 128 bits = 10라운드, 192 bits = 12라운드, 256 bits = 14라운드
    - Feistel 암호 구조가 아님
      - 각 라운드에서 대체와 치환을 이용 함

대칭 암호 알고리즘

- AES (Advanced Encryption Standard)
  - 암호화 과정
    - 키를 확장하여 각 라운드에 사용
    - 입력으로 들어온 평문을 4가지 단계로 암호화
  - 복호화 과정
    - 키를 확장하여 역순으로 사용
    - 입력으로 들어온 암호문을 4가지 단계로 복호화
    - 암호화 했던 4가지 단계 연산은 전부 역연산이 가능



대칭 암호 알고리즘

- AES (Advanced Encryption Standard)
  - 라운드 구조 (1/4)
    - 바이트 대체 (Substitute Bytes)
      - S-box를 이용하여 bits 단위인 블록을 Byte 단위로 변환

	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

< S-box >

19	A0	9A	E9
3D	F4	C6	F8
E3	E2	8D	48
BE	2B	2A	08

D4	E0	B8	1E
27	BF	B4	41
11	98	5D	52
AE	F1	E5	30

대칭 암호 알고리즘

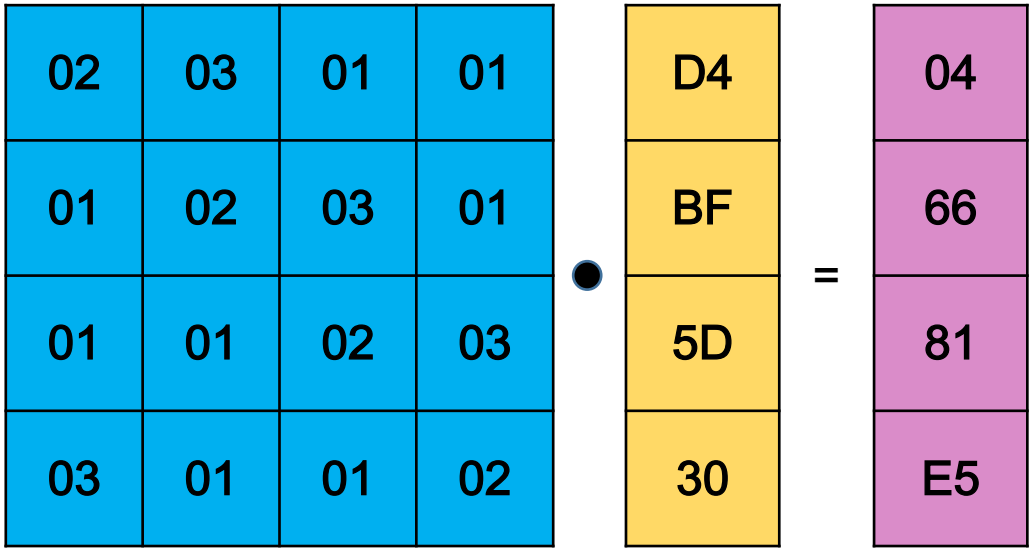
- AES (Advanced Encryption Standard)
  - 라운드 구조 (2/4)
    - 행 이동 (Shift rows)
      - 행과 행을 Byte 단위로 치환
        - 치환을 통해 암호화 과정 평문의 모든 비트에 영향을 주기 위함
        - 1행은 그대로
        - 2행은 Left shift 1회
        - 3행은 Left shift 2회
        - 4행은 Left shift 3회

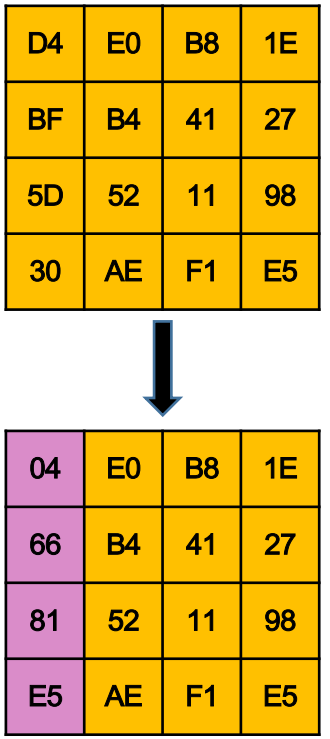
D4	E0	B8	1E
27	BF	B4	41
11	98	5D	52
AE	F1	E5	30

D4	E0	B8	1E
BF	B4	41	27
5D	52	11	98
30	AE	F1	E5

대칭 암호 알고리즘

- AES (Advanced Encryption Standard)
  - 라운드 구조 (3/4)
    - 열 섞기 (Mix columns)
      - 열에 있는 각 Byte를 대체하여 변환
      - 암호가 역으로 작동되기 위해 마지막 라운드에서는 수행하지 않음

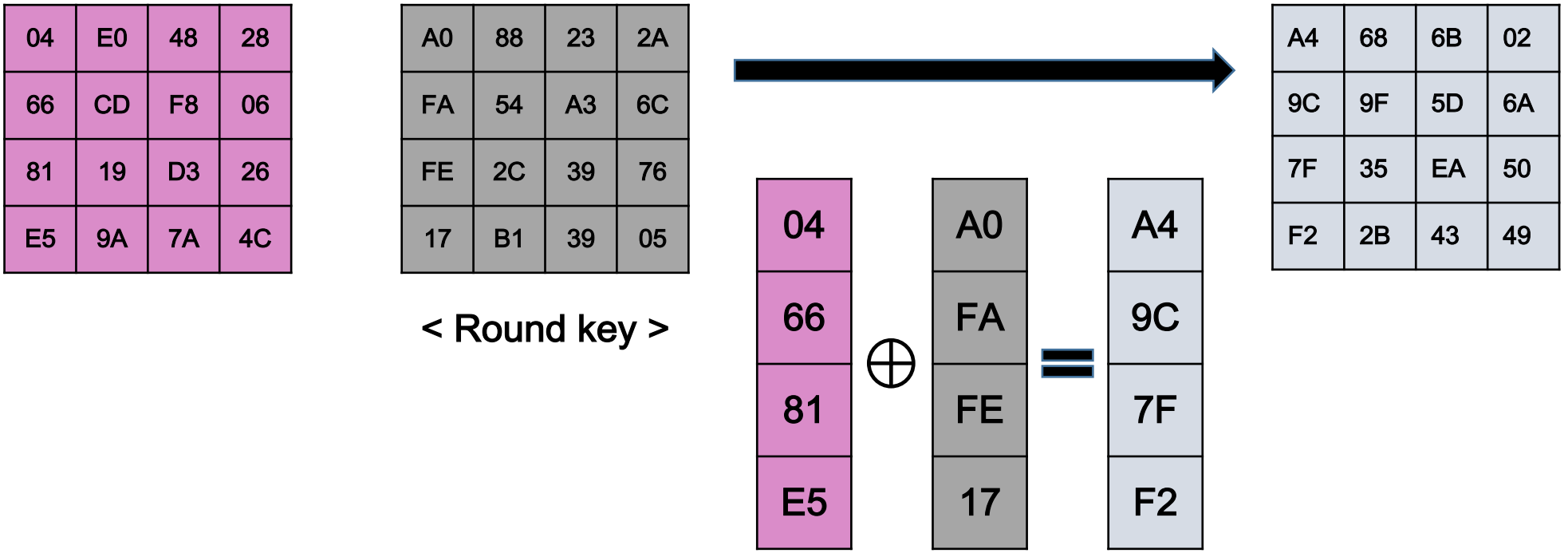




04	E0	48	28
66	CD	F8	06
81	19	D3	26
E5	9A	7A	4C

대칭 암호 알고리즘

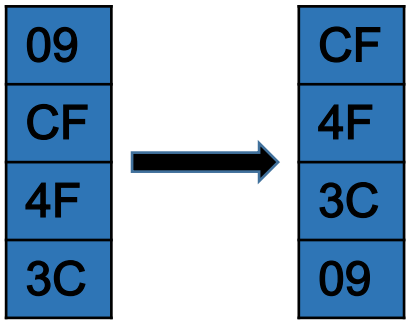
- AES (Advanced Encryption Standard)
  - 라운드 구조 (4/4)
    - 라운드 키 더하기 (Add round key)
      - 확장된 키와 현재 상태 배열에 있는 블록을 비트 별로 XOR연산



대칭 암호 알고리즘

- AES (Advanced Encryption Standard)
  - 키 확장 (Key schedule)
    - 키 확장을 통해 라운드 마다 사용되는 키 생성

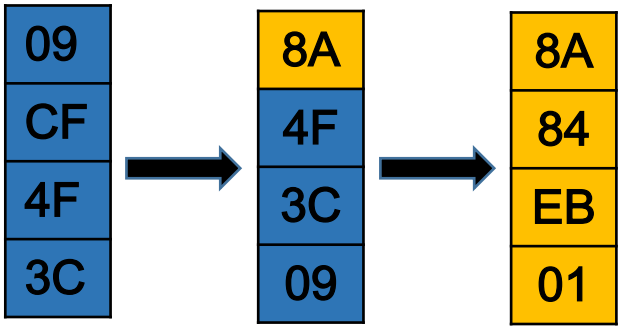
1. 열의 이동 (Shift column)



2. 바이트 대체 (Substitute bytes)

2B	28	AB	09
7E	AE	F7	CF
15	D2	15	4F
16	A6	88	3C

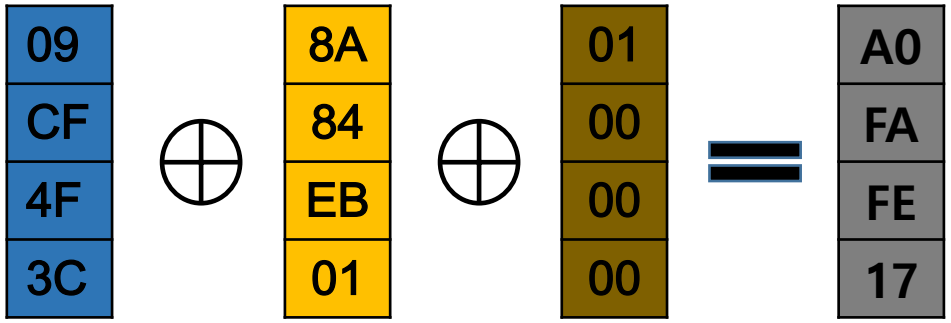
< **Cipher key** >



	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

대칭 암호 알고리즘

- AES (Advanced Encryption Standard)
  - 키 확장 (Key schedule)
    - 키 확장을 통해 라운드 마다 사용되는 키 생성
      1. XOR 연산
         - 새로 생성하는 라운드 키 행렬의 첫 번 째 열 생성
           - Cipher key의 첫 번째 열과 1,2번의 수행한 결과 값과 Rcon의 라운드 수 번째 열을 XOR연산

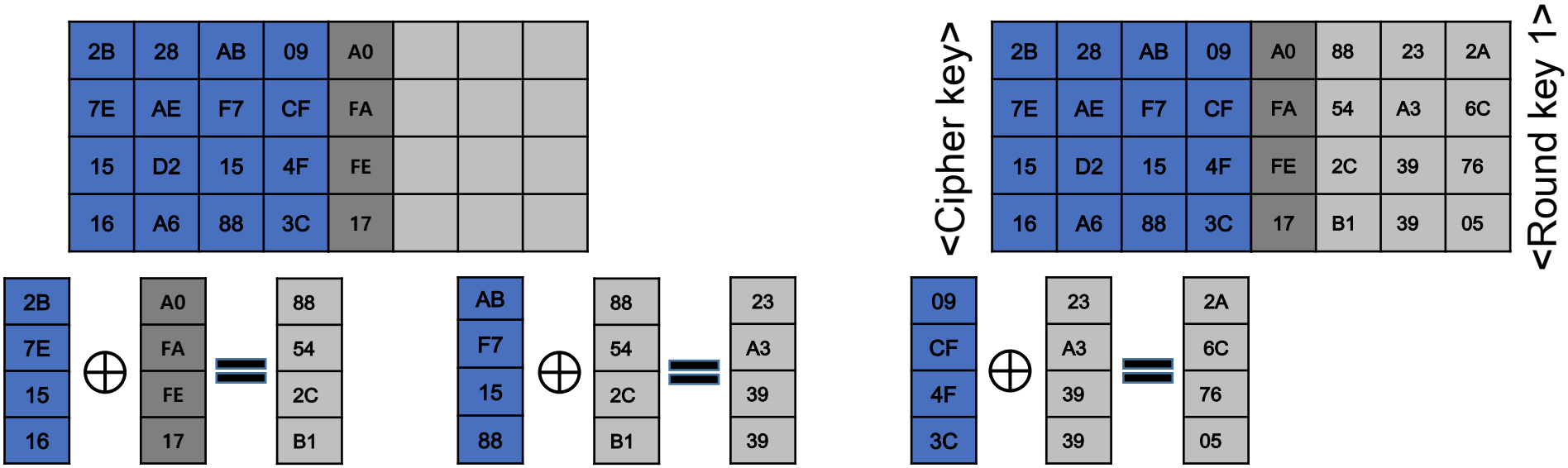


01	02	04	08	10	20	40	80	1B	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

< Rcon >

대칭 암호 알고리즘

- AES (Advanced Encryption Standard)
  - 키 확장 (Key schedule)
    - 키 확장을 통해 라운드 마다 사용되는 키 생성
      1. XOR 연산
         - 라운드 키의 2번째 열 계산
           - Rcon과 XOR연산한 값과 Cipher key의 첫번째 열을 XOR연산
         - 라운드 키의 3, 4 번째 열 계산
           - 새로운 행렬의 열과 기존 Cipher key의 열을 XOR연산



대칭 암호 알고리즘

- AES (Advanced Encryption Standard)
  - 키 확장 (Key schedule)
    - 생성된 라운드 키

Thanks!

박 재 형 (jaehyoung@pel.smuc.ac.kr)