2017/07/04, 2017 보안 기초 세미나

Network Security Essentials

- 2장 대칭 암호와 메시지 기밀성 (1) -

최창준 (changjun@pel.smuc.ac.kr)

상명대학교 프로토콜공학연구실

목 차

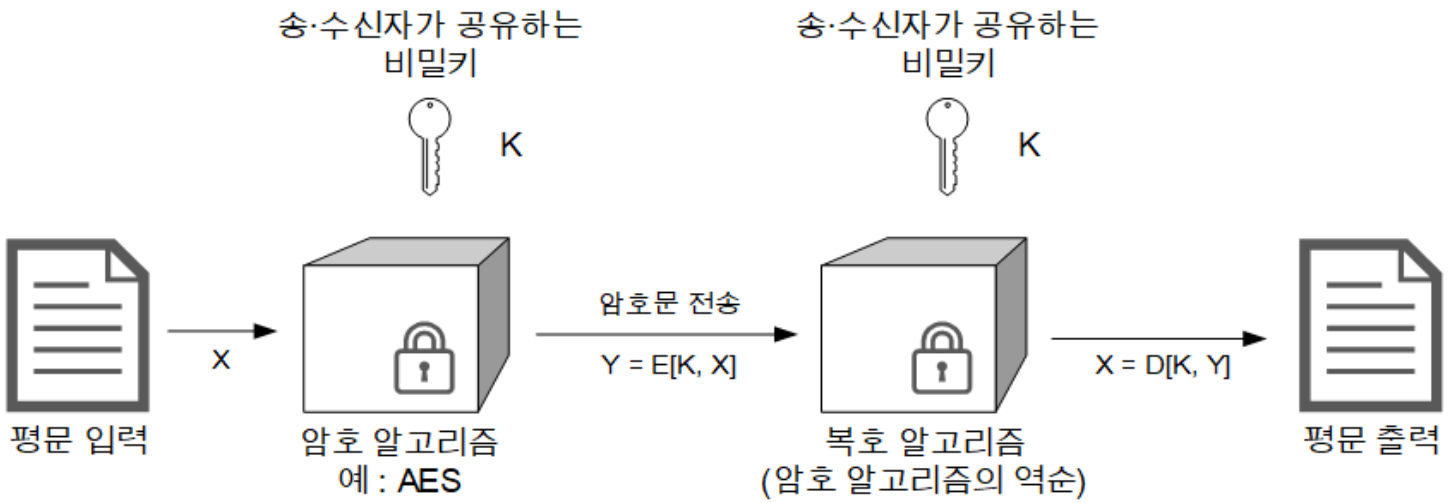
- 대칭 암호
  - 대칭 암호의 정의 및 구조
  - 암호 시스템의 3가지 독립적인 단계
  - 암호 공격 유형
  - Feistel 암호
- 대칭 암호 알고리즘
  - DES
  - 3DES
  - AES
- 랜덤넘버
  - 진성 랜덤 넘버
  - 의사 랜덤 넘버

대칭 암호

- 대칭 암호 (Symmetric Encryption)
  - 메시지를 암호화할 때 사용하는 키와 암호를 해독할 때 사용되는 키가 동일한 암호 방식
- 대칭 암호의 구조
  - 평문(Plaintext) : 원문 메시지 또는 데이터
  - 비밀키 (Secret Key) : 메시지를 암호화할 때 사용되는 키
  - 암호문 (Ciphertext) : 평문이 암호 알고리즘에 의해 암호화된 메시지
  - 암호 알고리즘 (Encryption Algorithm)
    - 평문을 암호화하여 암호문으로 변환하는 알고리즘
  - 복호 알고리즘 (Decryption Algorithm)
    - 암호문을 복호화하여 평문으로 복구해내는 알고리즘

대칭 암호

- 대칭 암호 단순 모델



- 송·수신자가 공유하는 키를 비밀로 하여 통신의 기밀성을 보장하는 기법
  - 키는 외부에 공개되지 않도록 안전하게 보관되어야 함
  - 알고리즘은 공개되어도 암호 해독에 전혀 도움이 되지 못함
    - 하지만 전체적인 알고리즘은 암호문을 쉽게 해독할 수 없도록 어렵게 만들어져야 함

대칭 암호

- 암호 시스템의 3가지 독립적인 단계 (1/2)
  - 평문을 암호문으로 전환하는 연산
    - 대체 (Substitution)
      - 각 요소(비트, 문자, 블록 등)를 다른 요소로 바꿔서 암호화
    - 치환 (Permutation)
      - 각 요소의 순서(위치)를 재조정하여 암호화
    - 대체와 치환 알고리즘을 같이 사용하면 하나의 알고리즘을 사용하여 생성된 암호문보다 더욱 해독하기 어려운 암호문을 생성할 수 있음
  - 사용되는 키
    - 대칭키 (비밀키)
      - 송·수신자 양측이 동일한 키를 사용
      - 비밀키로 암호화한 것을 같은 비밀키로 복호화

대칭 암호

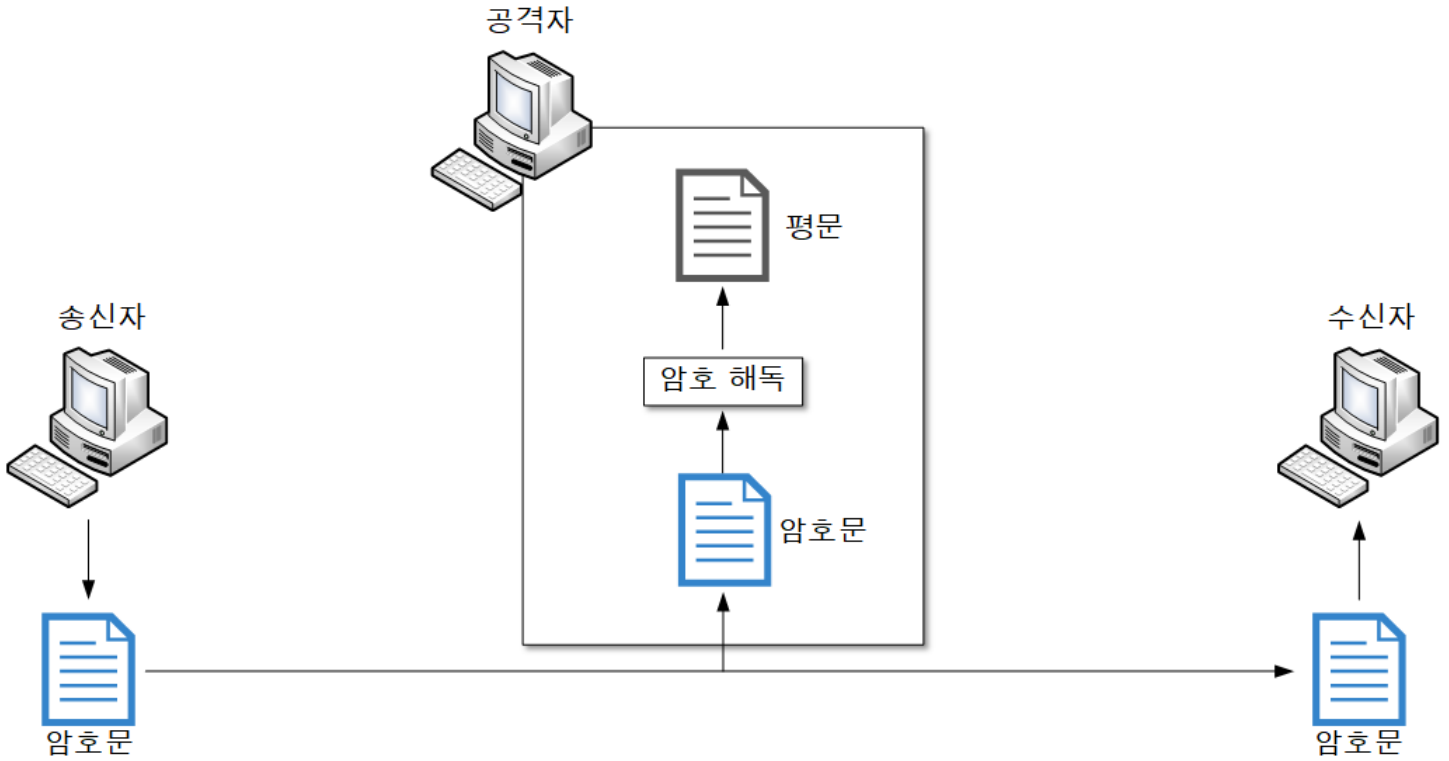
- 암호 시스템의 3가지 독립적인 단계 (2/2)
  - 대칭 암호의 분류
    - 블록 암호 (Block Cipher)
      - 평문 또는 기밀성 있는 정보를 정해진 블록 단위로 암·복호화
      - 한 번에 한 블록씩 입력하여 처리하고 한 블록씩 출력
      - e.g., 일반 데이터 전송 등
    - 스트림 암호 (Stream Cipher)
      - 평문 또는 기밀성 있는 정보를 비트 단위로 암·복호화
      - 연속적으로 처리하여 한 번에 한 요소씩 출력
      - e.g., 오디오/비디오 스트리밍 등

대칭 암호

- 암호 해독 (Cryptanalysis)
  - 암호문으로부터 평문 혹은 키를 찾으려는 시도
  - 평문의 성질과 암호에 관한 지식, 장비 등 여러 가지 정보를 이용하여 암호문으로부터 평문 혹은 키를 찾아냄

대칭 암호

- 암호 공격 유형 (1/5)
  - 암호문 단독 공격 (Ciphertext Only Attack) (1/2)
    - 암호문만을 가지고 평문의 성질, 문장의 특성 등을 추정하여 암호를 해독하는 공격 유형

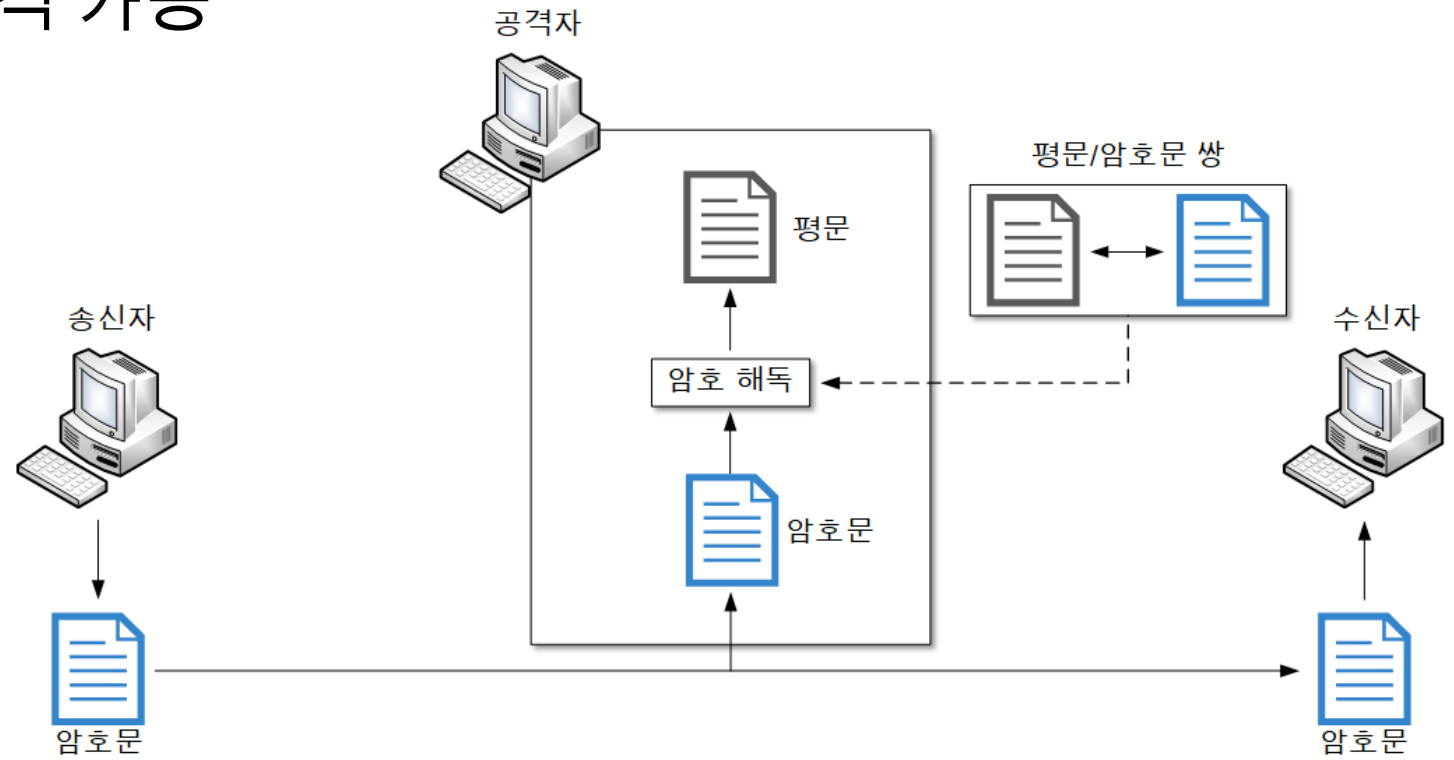


대칭 암호

- 암호 공격 유형 (2/5)
  - 암호문 단독 공격 (Ciphertext Only Attack) (2/2)
    - 암호문 단독 공격에서 사용되는 공격
      - 전수조사 공격 (Brute Force Attack)
        - 암호 해독에 있어 가능한 모든 경우를 전부 시도해보는 공격
      - 통계적 분석 공격 (Statistical Attack)
        - 암호문에서 통계적으로 많이 사용되는 평문 언어의 고유한 특징으로부터 정보를 얻어 수행하는 공격
        - e.g., 가장 빈번하게 사용되는 단어 또는 기호
      - 패턴 공격 (Pattern Attack)
        - 암호문에 존재하는 패턴을 이용하여 평문을 유추하는 공격

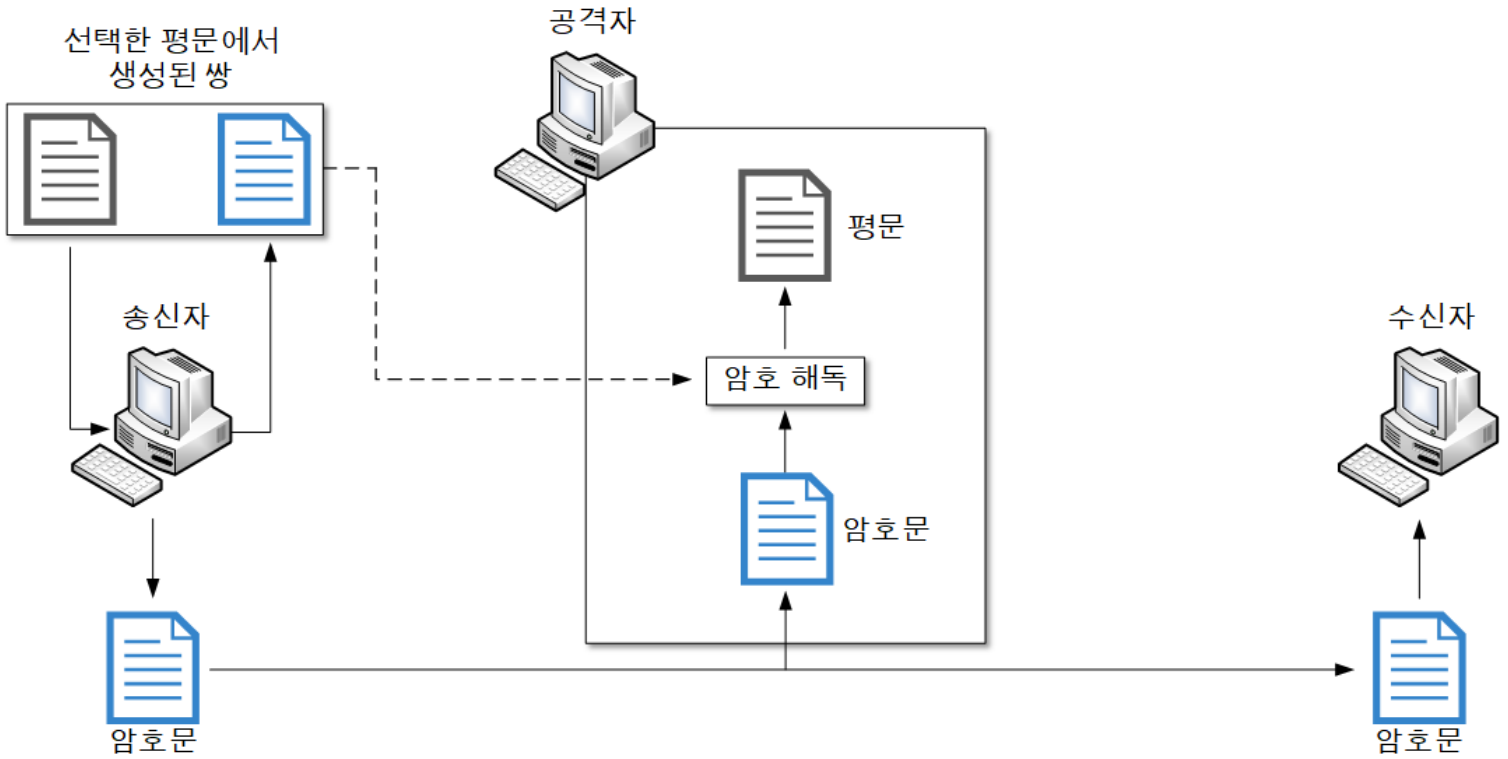
대칭 암호

- 암호 공격 유형 (3/5)
  - 알려진 평문 공격 (Known Plaintext Attack)
    - 평문, 암호문을 알고 있는 상태에서 암호문과 평문의 연관성을 추정하여 암호를 해독하는 공격 유형
    - 암호문 단독 공격과 마찬가지로 전수조사, 통계적 분석, 패턴 공격 가능



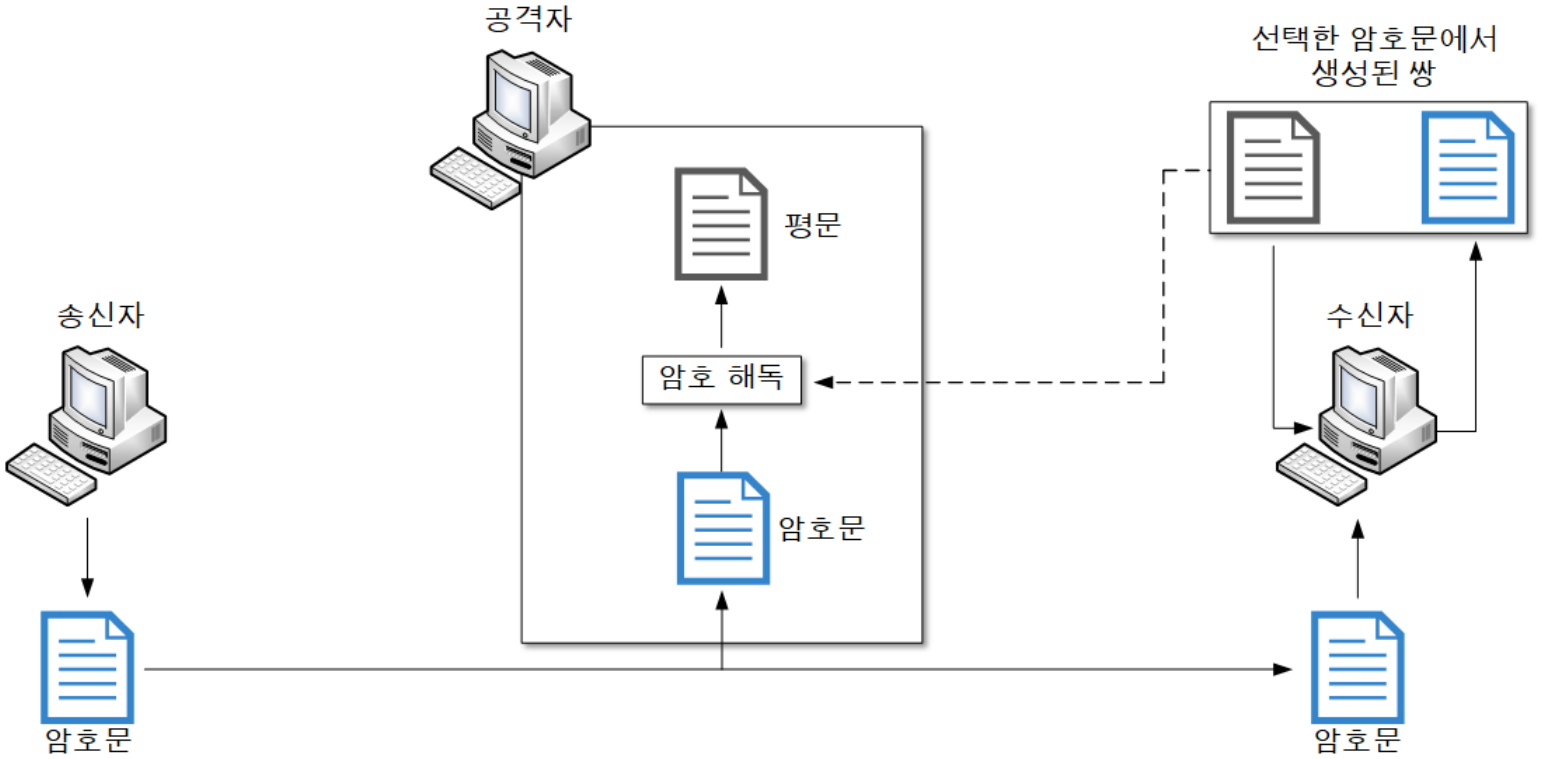
대칭 암호

- 암호 공격 유형 (4/5)
  - 선택 평문 공격 (Chosen Plaintext Attack)
    - 송신자의 컴퓨터에 접속하여 평문을 선택하고 그 평문에 해당하는 암호문을 얻어 암호를 해독하는 공격 유형



대칭 암호

- 암호 공격 유형 (5/5)
  - 선택 암호문 공격 (Chosen Ciphertext Attack)
    - 수신자의 컴퓨터에 접속하여 암호문을 선택하고 그 암호문에 해당하는 평문을 얻어 암호를 해독하는 공격 유형



대칭 암호

- Feistel 암호 (1/4)
  - 1973년 IBM의 독일 암호학자 Horst Feistel이 최초로 소개한 암호 구조
  - 특징
    - 대체와 치환을 번갈아 사용하여 메시지를 암·복호화
    - 64bits 블록 크기, 64bits 키 길이, 16 라운드 사용
    - 블록 크기와 키 길이가 길수록 안전성이 높음
      - 그 대신 암·복호화 속도가 떨어짐
    - 복호화 과정은 암호화 과정의 역순
    - 대부분의 대칭 블록 암호 알고리즘의 구조는 Feistel 암호 구조에 따라 만들어짐
      - e.g., DES, 3DES 등

대칭 암호

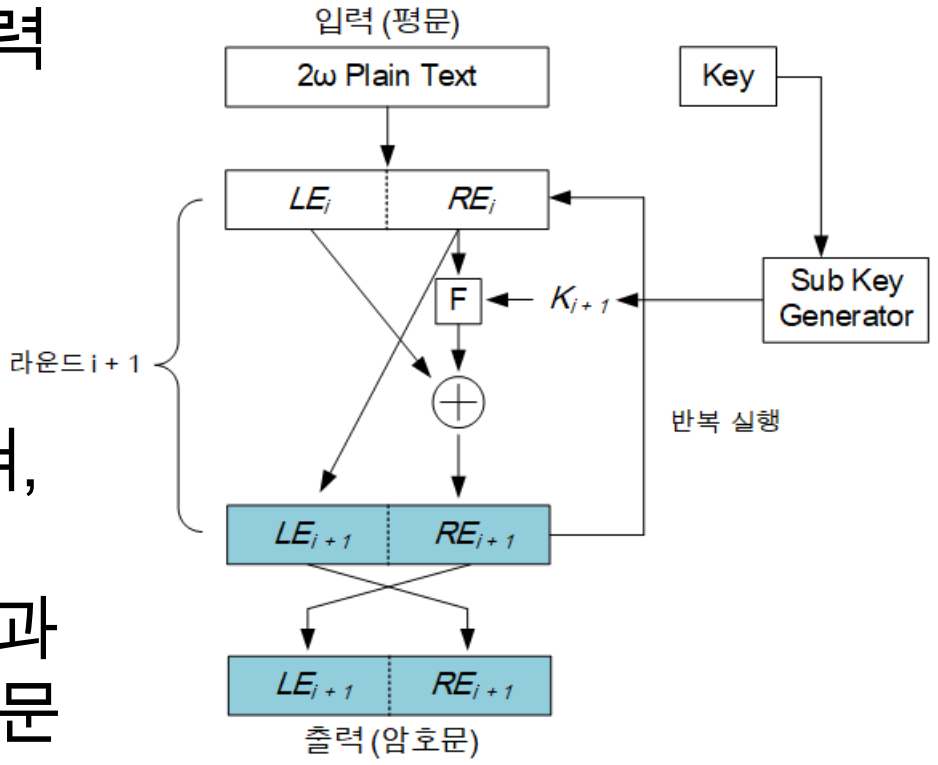
- Feistel 암호 (2/4)
  - 설계 특성
    - 라운드 함수 (Round Function)
      - 평문과 서브키를 입력 받아 대체와 치환을 수행하는 함수
    - 서브키 생성 알고리즘 (Sub-Key Generation Algorithm)
      - 키를 라운드 함수에서 입력되는 서브키로 변환하는 알고리즘

대칭 암호

- Feistel 암호 (3/4)
  - 설계할 때 고려해야 할 사항
    - 빠른 소프트웨어 암호/복호 (Fast Software En/Decryption)
      - 알고리즘의 실행속도를 고려하여 응용 프로그램이나 함수에 내장시킴
    - 용이한 해독 (Ease of Analysis)
      - 암호를 쉽게 해독하지 못하도록 알고리즘을 어렵게 만드는 것도 중요하지만 알고리즘의 구조를 단순하고 해독이 용이하게 설계하면 취약점을 쉽게 찾을 수 있고, 설계자는 그 취약점을 바탕으로 더 강한 보안성을 가지는 알고리즘을 만들 수 있음
  - 암호 구조가 계산적으로 안전한 구조
    - 암호문을 깨는 데 드는 비용이 암호화된 정보의 가치보다 큼
    - 암호문을 깨는 데 걸리는 시간이 해당 정보의 수명보다 김

대칭 암호

- Feistel 암호 (4/4)
  - 암호화 과정
    - 하나의 64bits 평문 블록을 입력
    - 평문 블록을 32bits씩 LE_i 과 RE_i으로 나눔
    - $RE_i = LE_{i+1}$
    - $f(RE_i, K_{i+1}) \oplus LE_i = RE_{i+1}$
    - 위의 과정이 1회의 라운드이며, 이 과정을 16회만큼 반복 실행
    - 마지막 라운드가 끝나면 LE_{16}과 RE_{16}의 위치를 교환하여 암호문 출력

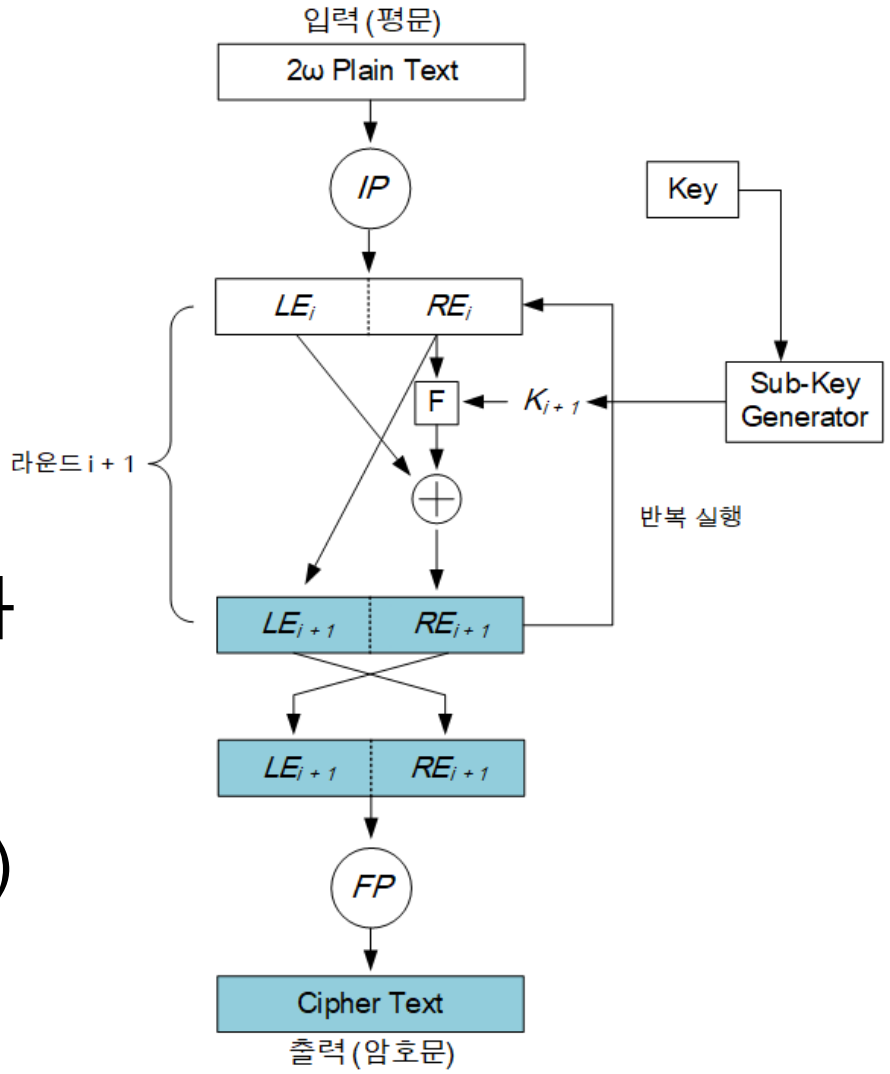


대칭 암호 알고리즘

- DES (Data Encryption Standard) (1/8)
  - 1972년 미국 국가기술표준원 (NIST : National Institute of Standards and Technology)이 개발한 미국 정부 규모의 표준적인 암호 알고리즘
  - 특징
    - 평문의 길이 64bits
    - 실제 키의 길이 56bits = 키의 길이 64bits - 패리티 비트 8bits
      - 패리티 비트 : 정보 전달 과정에서 오류를 검사하는 비트
    - 라운드 회수 16회
    - 길이가 56bits인 키로부터 길이가 48bits인 서로 다른 16개의 서브키 생성
    - 기본적인 Feistel 암호 알고리즘 특성을 가짐
    - 복호화 과정은 암호화 과정의 역순

대칭 암호 알고리즘

- DES (Data Encryption Standard) (2/8)
  - 암호화 과정 (1/2)
    - 입력으로 들어온 64bits 평문 블록은 초기 치환 *IP* 를 거쳐 LE_i 와 RE_i 으로 32bits씩 나뉨
    - $RE_i = LE_{i+1}$
    - $f(RE_i, K_{i+1}) \oplus LE_i = RE_{i+1}$
    - 위 과정을 16회 반복 실행
    - 마지막 라운드가 끝나면 LE_{16} 과 RE_{16} 의 위치 교환
    - 위치가 교환된 평문 블록들은 최종 치환 (초기 치환의 역치환) FP을 거쳐 64bits 암호문으로 출력됨



대칭 암호 알고리즘

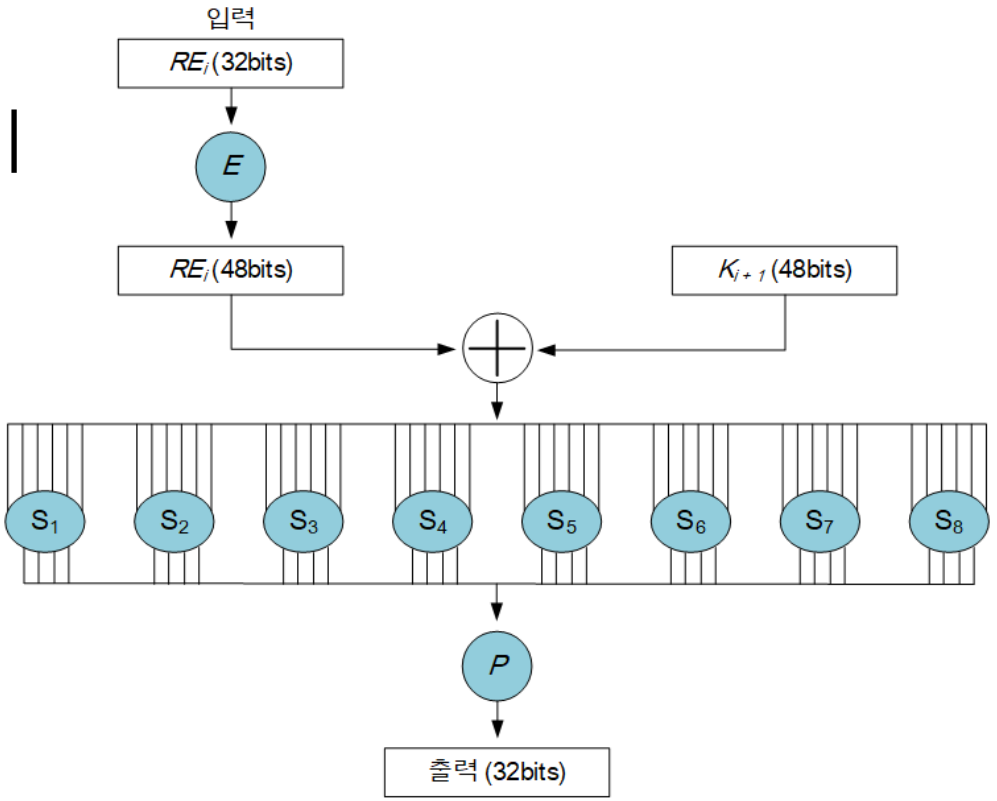
- DES (Data Encryption Standard) (3/8)
  - 암호화 과정 (2/2)
    - 초기 치환 (Initial Permutation)
      - 64bits를 입력 받아 정의된 규칙에 따라 재배열
    - 최종 치환 (Final Permutation)
      - 초기 치환의 역 관계

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 (Data Encryption Standard) (4/8)
  - 라운드 함수 (Round Function) (1/3)
    - 입력으로 들어온 32bit 오른쪽 평문 블록 RE_i은 확장 치환 E를 거쳐 48bit로 확장됨
    - 48bit로 확대된 RE_i은 서브키 K_{i+1}와 XOR연산 후 8개의 S-Box에 6bit씩 입력됨
    - S-Box에서 4bits씩 축소되어 출력된 비트들의 총 합 32bits는 P-Box를 통해 함수에서 출력됨



대칭 암호 알고리즘

- DES (Data Encryption Standard) (5/8)
  - 라운드 함수 (Round Function) (2/3)
    - 확장 치환 (Expansion Permutation)
      - 32bits를 입력 받아 정의된 규칙에 따라 48bits로 확장

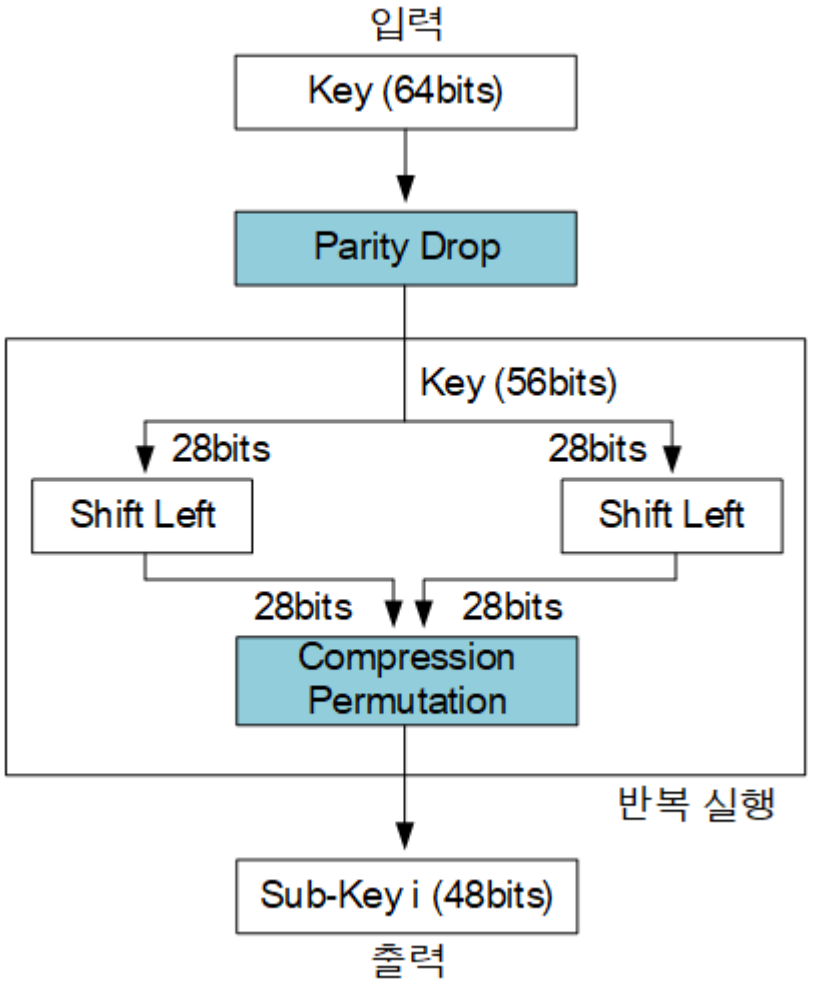
확장 치환					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

대칭 암호 알고리즘

- DES (Data Encryption Standard) (6/8)
  - 라운드 함수 (Round Function) (3/3)
    - S-Box (Substitution-Box)
      - 6bits의 입력을 4bits의 출력으로 축소시켜 변환하는 함수
      - 역방향으로 복원이 어려움
      - 1bit 입력이 2bits 이상의 출력을 나타내어야 함
    - P-Box (Permutation-Box)
      - S-Box 입력을 통해 bit를 치환하여 출력하는 함수

대칭 암호 알고리즘

- DES (Data Encryption Standard) (7/8)
  - 서브키 생성기 (Sub-Key Generator) (1/2)
    - 입력으로 들어온 64bits 키는 Parity Drop을 거쳐 키 생성 과정 전에 Parity bits (8bits)를 제거
    - 56bits 키를 28bits씩 나눠 비트를 좌측으로 순환 이동시킴
      - 1, 2, 9, 16 라운드는 1bit씩
      - 나머지 라운드는 2bits씩
    - 축소 치환을 통해 56bits 키를 48bits로 축소하여 서브키로 출력
    - 위 과정을 16회 반복 실행



대칭 암호 알고리즘

- DES (Data Encryption Standard) (8/8)
  - 서브키 생성기 (Sub-Key Generator) (2/2)
    - 좌측 순환 이동 (Shift Left)

라운드	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits 이동	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

    - 축소 치환 (Compression Permutation)
      - 56bits를 입력 받아 정의된 규칙에 따라 48bits로 축소

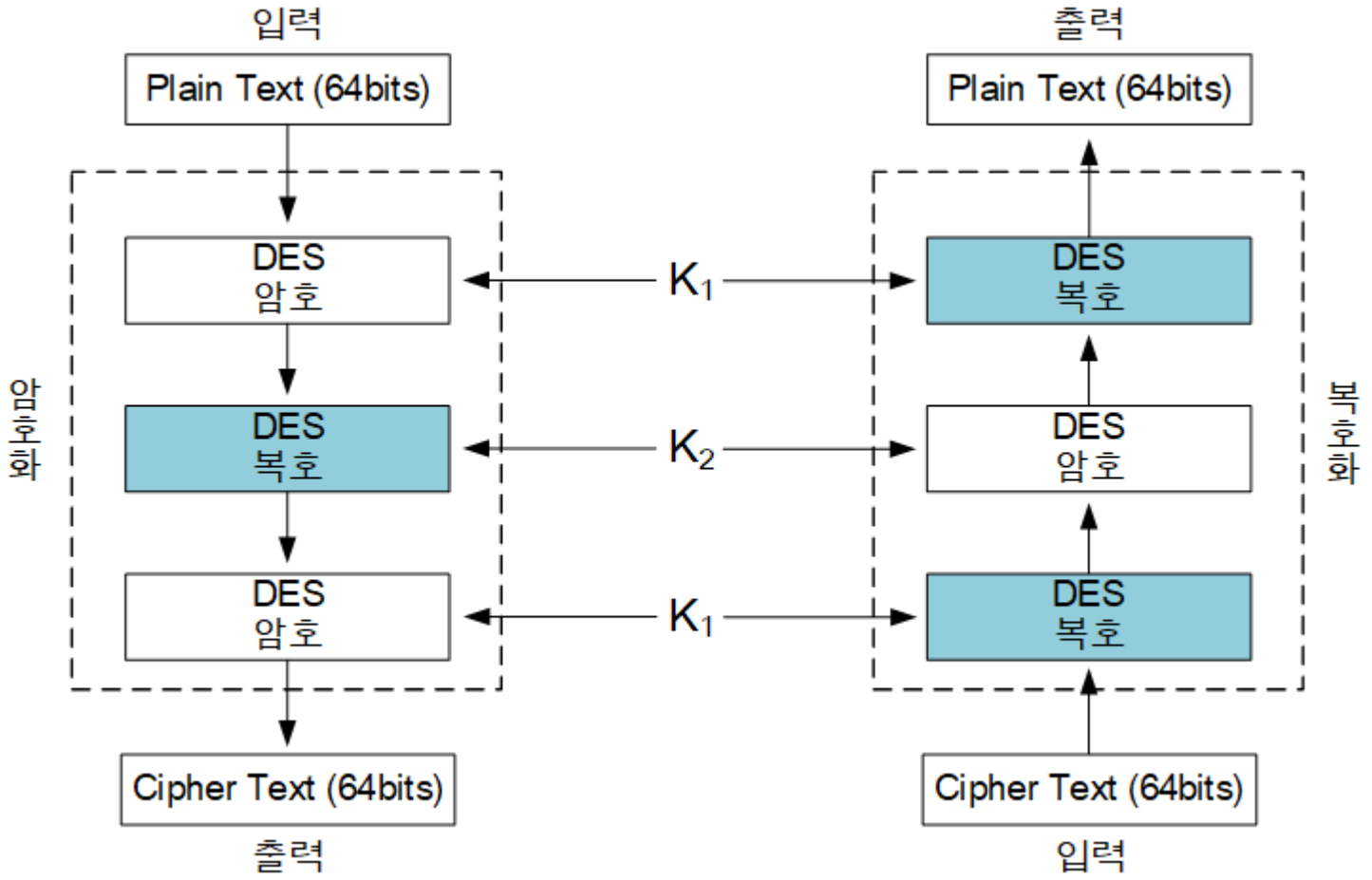
축소 치환							
14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

대칭 암호 알고리즘

- 3DES (Triple Data Encryption Standard) (1/3)
  - DES의 안전성을 향상시키기 위해 암호화와 복호화에 대하여 DES를 세 번 사용한 알고리즘
  - 특징
    - 2개의 키를 갖는 3DES
      - 길이가 56bits인 키를 2개 사용하여 DES의 암·복호화를 3회 혼용
      - 알려진 평문 공격에 취약
    - 3개의 키를 갖는 3DES
      - 길이가 56bits인 키를 3개 사용하여 DES의 암·복호화를 3회 혼용
      - 2개의 키를 갖는 3DES의 취약점을 개선
    - 보안성이 강화되어 전수 공격에 대한 DES의 취약점을 극복
    - 라운드가 3배가 되어 속도가 느림

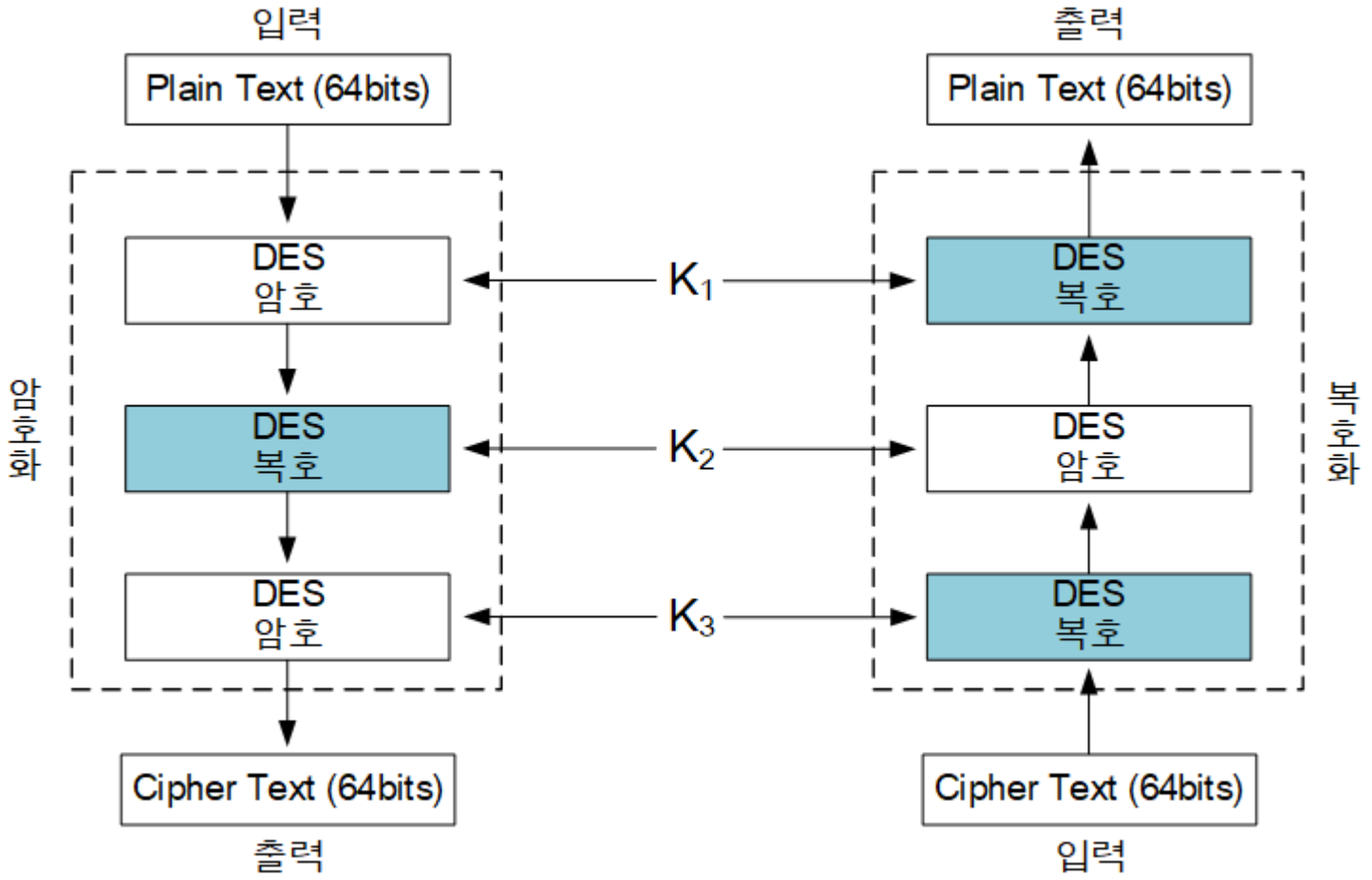
대칭 암호 알고리즘

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



대칭 암호 알고리즘

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

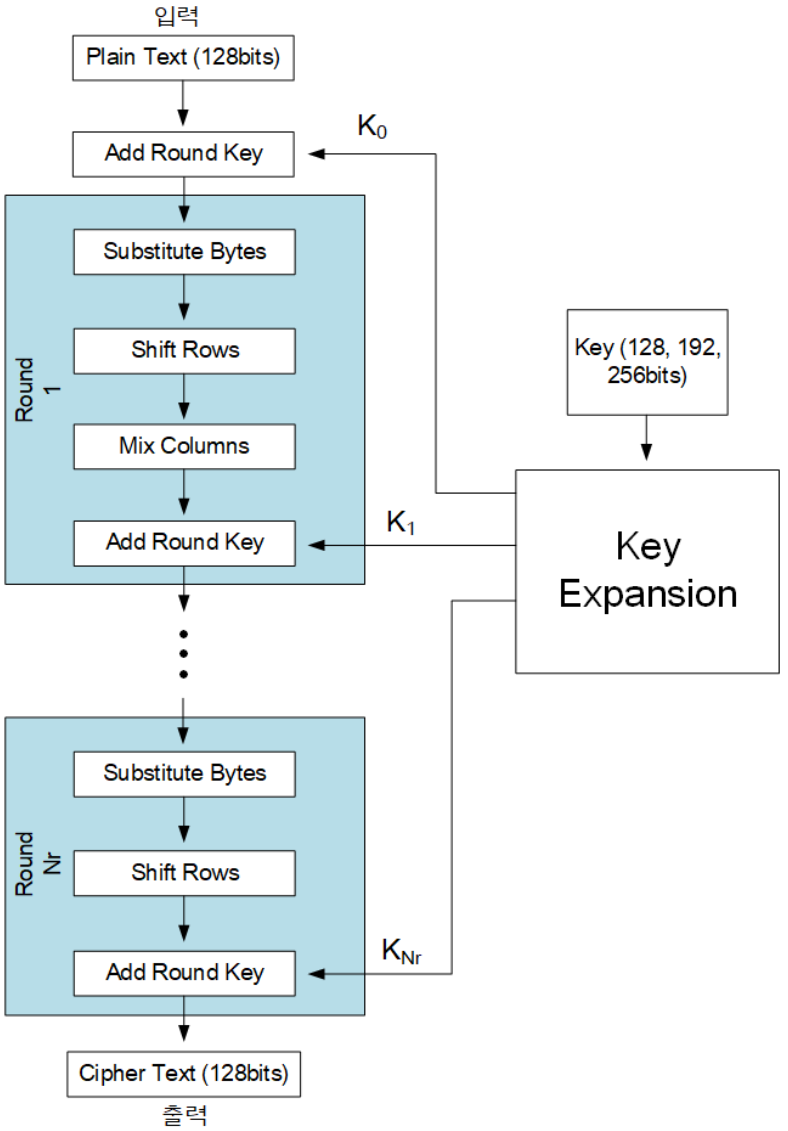


대칭 암호 알고리즘

- AES (Advanced Encryption Standard) (1/11)
  - 2001년 미국 국립기술표준원 (NIST : National Institute of Standards and Technology)에서 공표한 대칭키 암호 알고리즘
  - 두 명의 벨기에 암호학자 Joan Deamen과 Vincent Rijmen이 개발한 Rijndeal 블록 암호 알고리즘
  - 특징
    - 128bits 블록과 길이가 128, 192, 256bits인 키를 사용
      - 128bits – 10 라운드, 192bits – 12 라운드, 256bits – 14 라운드
    - 기본적인 SPN 암호 알고리즘 특성을 가짐
      - S-Box, P-Box를 이용하여 비트 이동 없이 한 번에 암·복호화 함
    - 복호화 과정은 암호화 과정의 역순

대칭 암호 알고리즘

- AES (Advanced Encryption Standard) (2/11)
  - 암호화 과정
    - 키를 확장하여 각 라운드에 사용
    - 입력으로 들어온 평문을 4가지 단계로 암호화
  - 복호화 과정
    - 키를 확장하여 역순으로 사용
    - 암호화했던 4가지 단계 연산은 전부 역연산이 가능
      - 입력으로 들어온 암호문을 4가지 단계로 복호화



대칭 암호 알고리즘

- AES (Advanced Encryption Standard) (3/11)
  - 라운드 구조 (1/4)
    - 바이트 대체 (Substitute Bytes)
      - S-Box를 이용하여 bits 단위인 블록을 Byte 단위로 변환

hex	y															
x	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

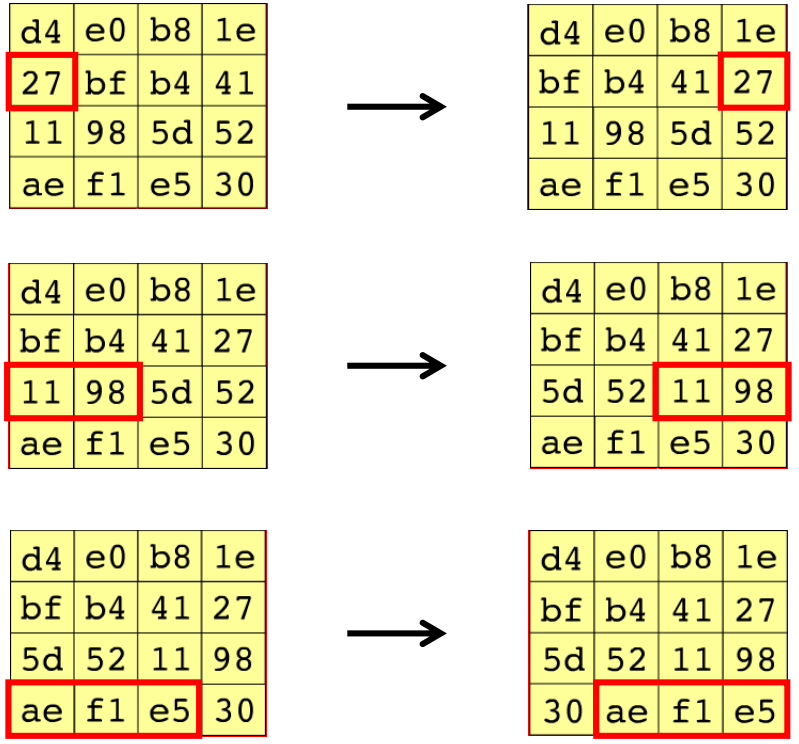
S-Box

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

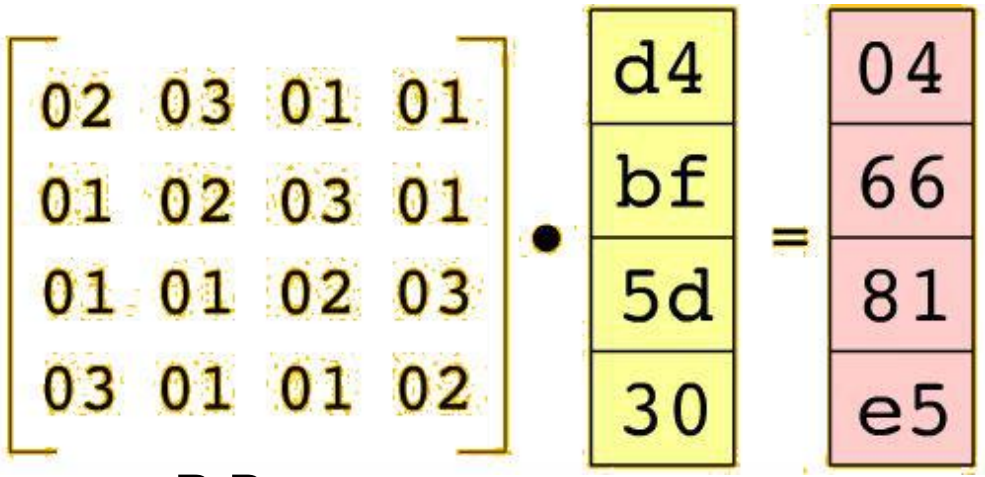
대칭 암호 알고리즘

- AES (Advanced Encryption Standard) (4/11)
  - 라운드 구조 (2/4)
    - 행 이동 (Shift Rows)
      - 행과 행을 Byte 단위로 치환
        - 치환을 통해 암호화 과정 평문의 모든 비트에 영향을 주기 위함

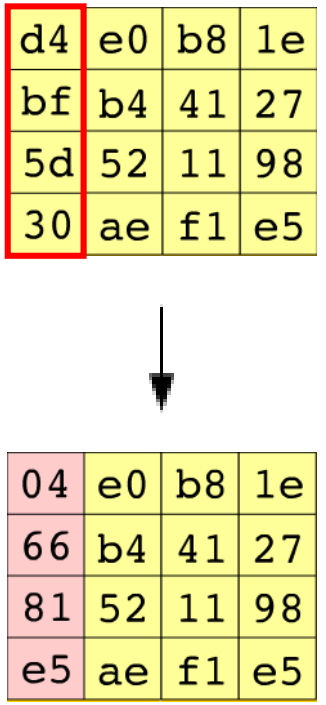


대칭 암호 알고리즘

- AES (Advanced Encryption Standard) (5/11)
  - 라운드 구조 (3/4)
    - 열 섞기 (Mix Columns)
      - 열에 있는 각 Byte를 대체하여 변환
      - 암호가 역으로 작동되기 위해 마지막 라운드에서는 수행하지 않음



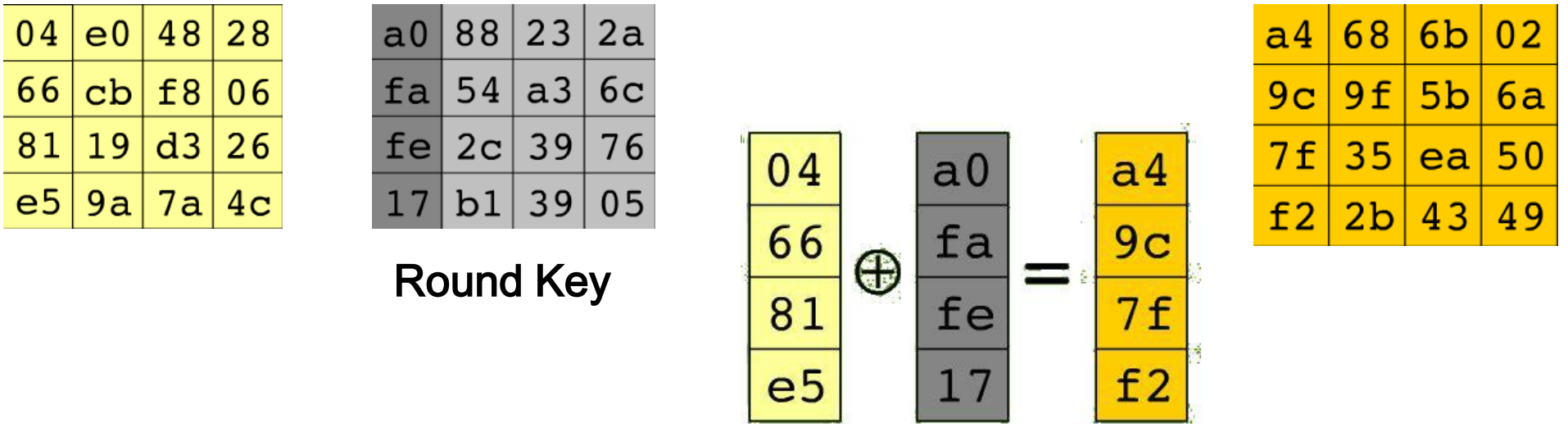
P-Box



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

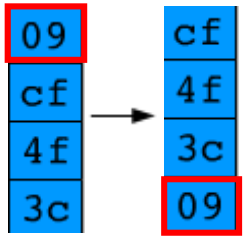
대칭 암호 알고리즘

- AES (Advanced Encryption Standard) (6/11)
  - 라운드 구조 (4/4)
    - 라운드 키 더하기 (Add Round Key)
      - 확장된 키와 현재 상태 배열에 있는 블록을 비트별로 XOR연산
      - 1 라운드를 수행하기 전에 초기 평문과 라운드 키를 XOR하는 과정이 필요하기 때문에 처음 라운드에 들어오기 전에 1번 수행

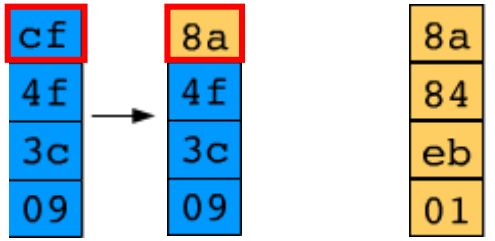


대칭 암호 알고리즘

- AES (Advanced Encryption Standard) (7/11)
  - 키 확장(Key Schedule) (1/4)
    - 키를 확장을 통해 라운드 마다 사용되는 키 생성
      1. 열 이동(Shift Column)



      2. 바이트 대체(Substitute Bytes)

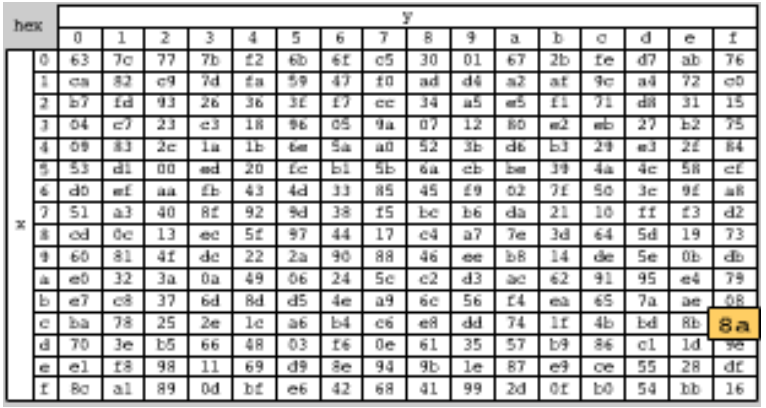


Cipher key

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

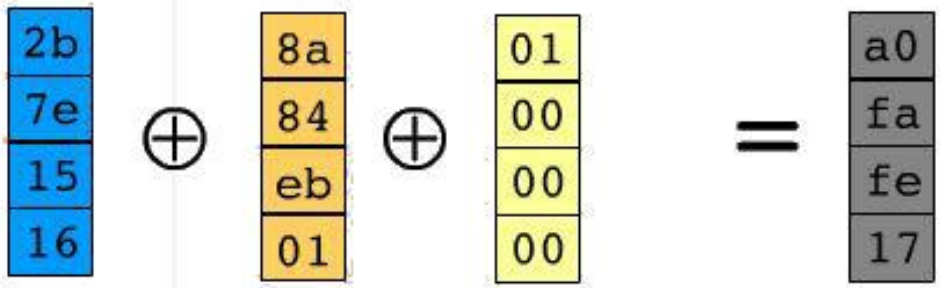
Rcon

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 (Advanced Encryption Standard) (8/11)
  - 키 확장(Key Schedule) (2/4)
    - 키를 확장을 통해 라운드 마다 사용되는 키 생성
      3. XOR 연산
         - 새로 생성하는 라운드 키 행렬의 첫 번 째 열 생성
           - Cipher key의 첫 번째 열과 1,2번을 수행한 결과 값과 Rcon의 라운드 수 번째 열을 XOR 연산



Cipher key

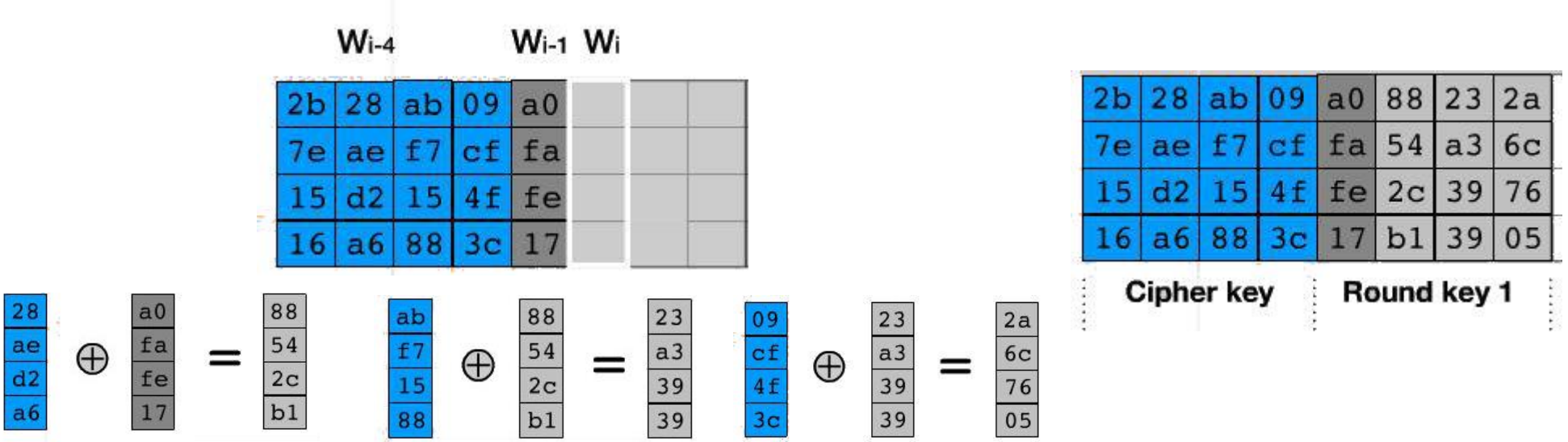
2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

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

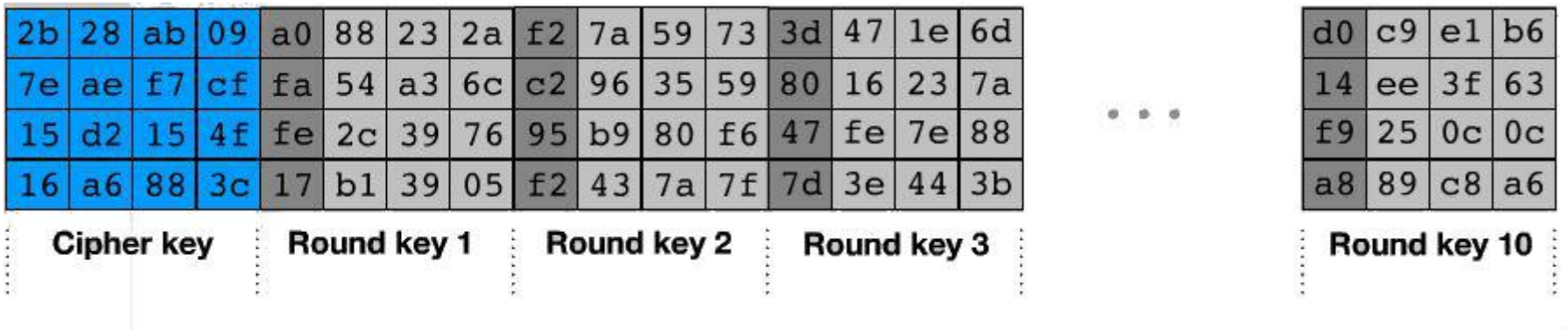
대칭 암호 알고리즘

- AES (Advanced Encryption Standard) (9/11)
  - 키 확장(Key Schedule) (3/4)
    - 키를 확장을 통해 라운드 마다 사용되는 키 생성
      3. XOR 연산
         - 라운드 키의 2번 째 열 계산
           - Rcon과 XOR 연산한 값(새로 생성한 행렬의 첫 번째 열)과 Cipher key의 첫 번째 열을 XOR 연산
         - 라운드 키 의 3, 4 번째 열 계산
           - 새로운 행렬의 열과 기존 Cipher key의 열을 XOR 연산



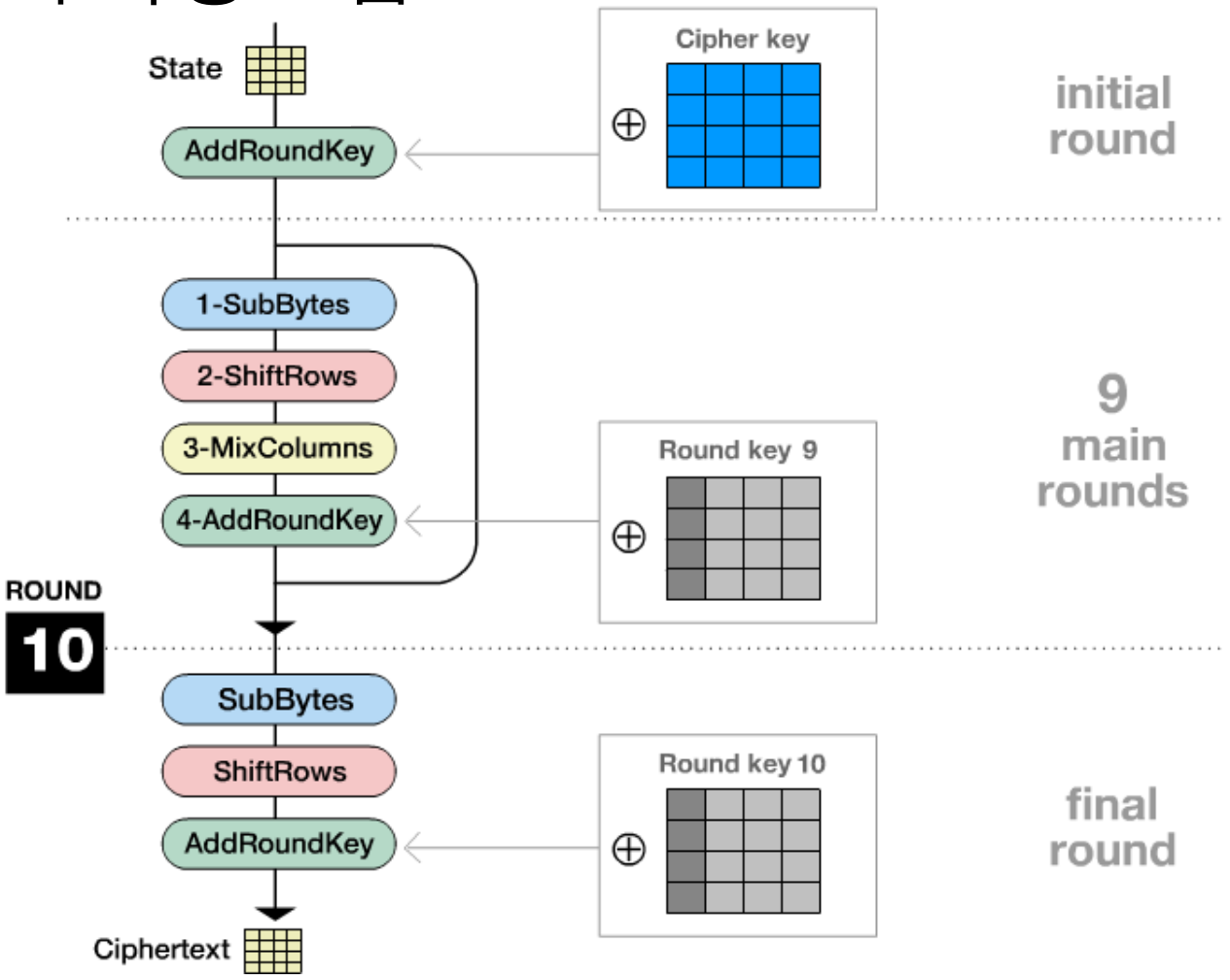
대칭 암호 알고리즘

- AES (Advanced Encryption Standard) (10/11)
  - 키 확장(Key Schedule) (4/4)
    - 생성된 라운드 키



대칭 암호 알고리즘

- AES (Advanced Encryption Standard) (11/11)
  - 전체 동작 과정 그림

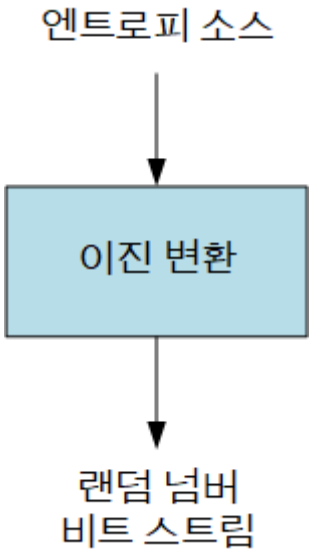


랜덤넘버

- 랜덤 넘버 (Random Number) (1/4)
  - 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 추출한 부분 수열이 다른 수열과 연관성이 없어야 함
    - 예측 불가능성 (Unpredictability)
      - 수열의 일부를 보고 다음에 이어지는 수를 예측할 수 없어야 함

랜덤넘버

- 랜덤 넘버 (Random Number) (2/4)
  - 진성 랜덤 넘버 (True Random Number)
    - 입력 값이 실제로 랜덤한 정보
    - 컴퓨터에서 물리적으로 얻을 수 있는 랜덤한 정보
      - 엔트로피 소스 (Entropy Source)
        - e.g., 키 입력 타이밍 패턴, 마우스 움직임, 시스템 클록의 순간 값 등
  - TRNG (True Random Number Generator)
    - 입력으로 들어온 엔트로피 소스의 조합을 이진 변환하여 랜덤한 바이너리를 출력하는 알고리즘

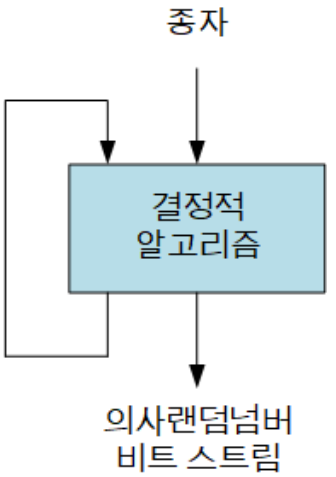


랜덤넘버

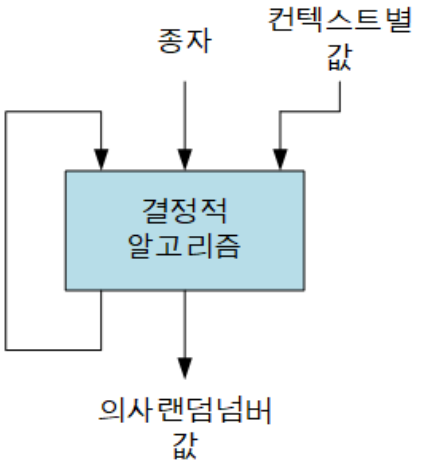
- 랜덤 넘버 (Random Number) (3/4)
  - 의사 랜덤 넘버 (Pseudo Random Number) (1/2)
    - 입력되는 종자 (Seed) 값에 따라 난수가 결정됨
      - 종자 : 고정된 값
    - 결정적 알고리즘 사용
      - 입력 값(종자 값)이 같으면 출력 값이 같은 알고리즘
      - 출력 값을 다시 입력 값으로 사용하기도 함

랜덤넘버

- 랜덤 넘버 (Random Number) (4/4)
  - 의사 랜덤 넘버 (Pseudo Random Number) (2/2)
    - PRNG (Pseudo Random Number Generator)
      - 무한 비트열을 생성하기 위해 사용되는 알고리즘



    - PRF (Pseudo Random Function)
      - 고정된 길이의 의사 랜덤 비트열을 생성하기 위해 사용되는 함수
        - 컨텍스트별 값 : 상황에 따른 입력값
      - 생성되는 비트열만 다를 뿐 PRNG와 차이점이 없음



감사합니다!