

# Network Security Essentials

## - Chapter\_2 대칭 암호와 메시지 기밀성(1) -

김 지 혜([jihye@pel.sejong.ac.kr](mailto:jihye@pel.sejong.ac.kr))

세종대학교 프로토콜공학연구실

# 목 차

---

- 보충
  - 컴퓨터 보안 개념
  - OSI 보안 구조
- 대칭 암호 원리
- 대칭 암호 알고리즘
- 랜덤 넘버와 의사 랜덤 넘버

# 목 차

---

- 보충

- 컴퓨터 보안 개념
- OSI 보안 구조
- 대칭 암호 원리
- 대칭 암호 알고리즘
- 랜덤 넘버와 의사 랜덤 넘버

# 보충

- 컴퓨터 보안 개념
  - 핵심 보안 요소

보안 요소		정의	설명
CIA Triad	기밀성 (Confidentiality)	<ul style="list-style-type: none"><li>• 인가된 사용자만 정보 접근이 가능해야 하는 것을 의미</li></ul>	<ul style="list-style-type: none"><li>• 개인 및 기밀 정보가 유출되는 경우, 기밀성 침해</li></ul>
	무결성 (Integrity)	<ul style="list-style-type: none"><li>• 정보가 원본을 유지해야 하는 것을 의미</li></ul>	<ul style="list-style-type: none"><li>• 정보가 위조/변조되거나 삭제되는 경우, 무결성 침해</li></ul>
	가용성 (Availability)	<ul style="list-style-type: none"><li>• 필요 시, 인가된 사용자의 접근을 가능하게 하는 것을 의미</li></ul>	<ul style="list-style-type: none"><li>• 강력한 보안으로 인해 권한이 부여된 사용자가 시스템을 사용하지 못하는 경우, 가용성 침해</li></ul>
Triple -A	인증 (Authentication)	<ul style="list-style-type: none"><li>• 접근을 위해 사용자의 신원을 검증하는 것을 의미</li></ul>	<ul style="list-style-type: none"><li>• 자신의 신원을 시스템에 증명하는 것으로, ID/PW를 입력하는 과정</li></ul>
	권한부여 (Authorization)	<ul style="list-style-type: none"><li>• 인증된 사용자의 서비스 허용 정도를 결정하는 것을 의미</li></ul>	<ul style="list-style-type: none"><li>• 지문이나 PW 등을 통해 로그인이 허락된 사용자로 판명되어 로그인하는 과정</li></ul>
	계정관리 (Accounting)	<ul style="list-style-type: none"><li>• 사용자 자원 정보를 측정하는 것을 의미</li></ul>	<ul style="list-style-type: none"><li>• 로그인 시, 시스템에 이에 대한 기록을 남기는 과정</li></ul>

# 보충

## • OSI 보안 구조

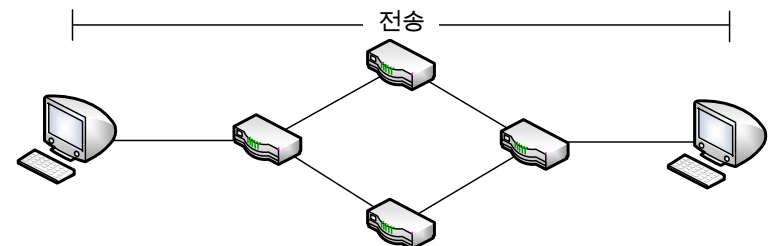
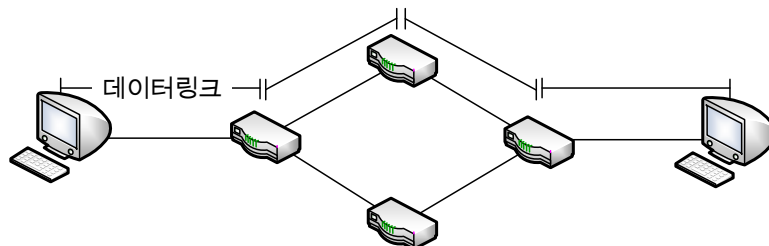
### • OSI(Open Systems Interconnection)

#### • 정의

- 시스템 간 원활한 통신을 위해 국제 표준화 기구에서 개발한 모델

#### • 계층 구조

계층		역할
1	물리(Physical)	• 장비에 직접 연결되어 데이터 전송
2	데이터 링크(Data Link)	• 데이터의 안전한 전달을 위한 오류 탐지 및 흐름 제어(단일 링크 수행, 송신 장비의 속도만 고려)
3	네트워크(Network)	• IP 주소 부여 및 라우팅 경로 지정
4	전송(Transport)	• 패킷 처리 및 전송, 혼잡 및 흐름 제어(종단-대-종단 수행, 경로에 따른 전 장비의 속도 고려)
5	세션(Session)	• 장비 간 통신을 위한 세션 설정
6	표현(Presentation)	• 응용 프로그램을 위한 데이터 표현
7	응용(Application)	• 사용자와 직접 상호작용



# 목 차

---

- 보충
  - 컴퓨터 보안 개념
  - OSI 보안 구조
- 대칭 암호 원리
- 대칭 암호 알고리즘
- 랜덤 넘버와 의사 랜덤 넘버

# 대칭 암호 원리

- 암호

- 송수신자가 안전하지 않은 채널을 통해 정보를 주고받더라도 제3자는 이 정보의 내용을 알 수 없도록 하는 것

- 암호화 방식(1/3)

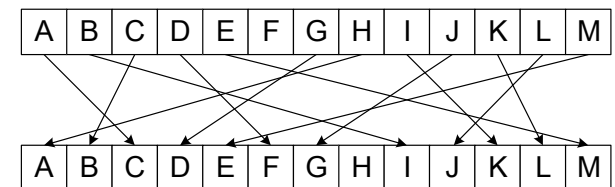
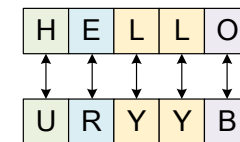
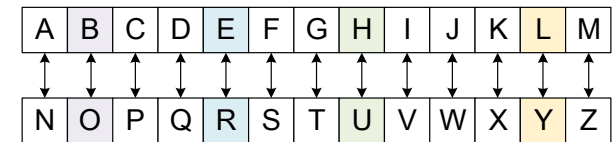
- 암호화에 사용되는 연산 유형

- 치환 암호(Substitution Cipher)

- 문자를 다른 문자로 대체하는 암호 방식

- 전치 암호(Transposition Cipher)

- 문자들의 순서를 일대일로 재조정하는 암호 방식



# 대칭 암호 원리

- 암호화 방식(2/3)

- 사용되는 키의 수에 따른 암호 방식

- 대칭 암호(Symmetric Encryption)

- 송수신자가 동일한 키를 가지고 암호화/복호화하는 방식

- 비대칭 암호(Asymmetric Encryption)

- 송수신자가 서로 다른 키를 가지고 암호화/복호화하는 방식

특징	대칭 암호	비대칭 암호
속도	상대적으로 빠름	상대적으로 느림
키 관계	암호 키 = 복호 키	암호 키 $\neq$ 복호 키
키 전송	필요	불필요
키 길이	키 길이가 짧음	키 길이가 김
대표 알고리즘	e.g., DES, AES	e.g., Diffie-Hellman, RSA



# 대칭 암호 원리

- 암호화 방식(3/3)

- 평문 처리 방법에 따른 암호 방식

- 블록 암호(Block Cipher)

- 평문을 블록 단위로 나누어서 암호화/복호화하는 방식

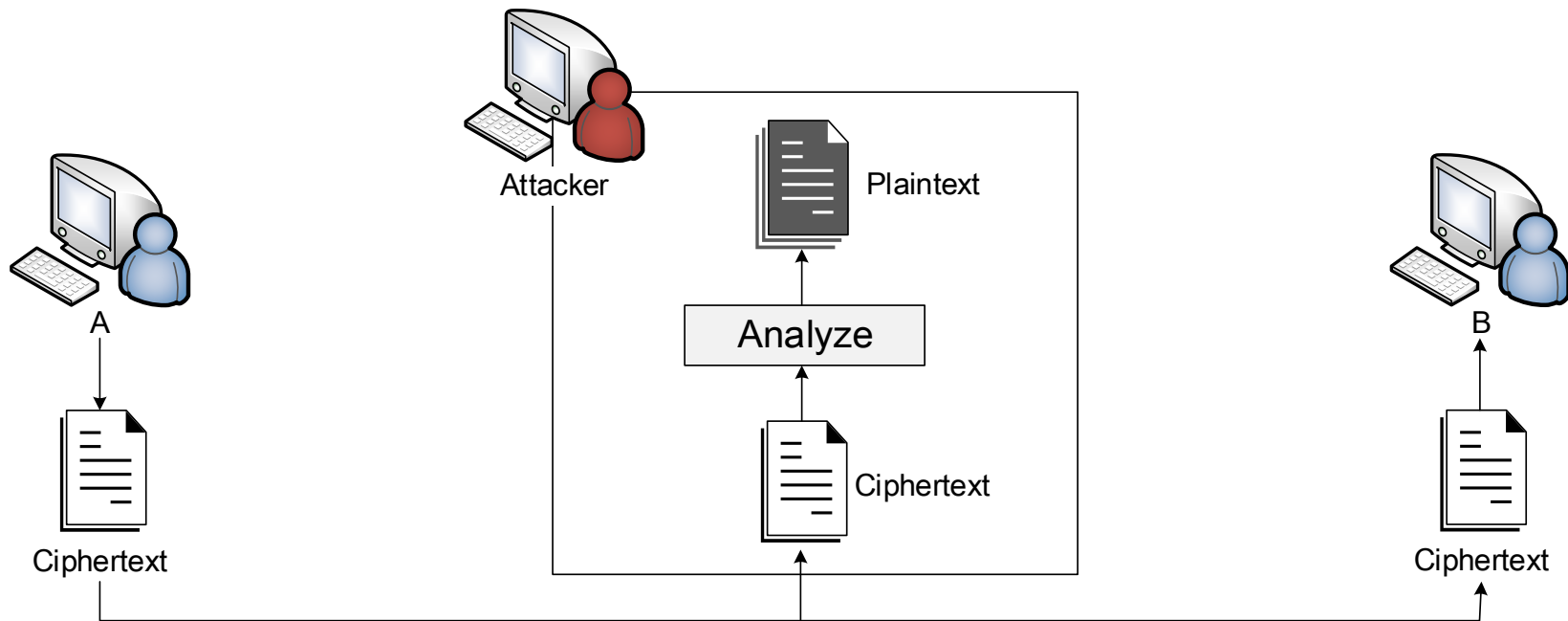
- 스트림 암호(Stream Cipher)

- 연속적인 비트 혹은 바이트를 순차적으로 암호화/복호화하는 방식

특징	블록 암호	스트림 암호
단위	블록	비트
속도	상대적으로 느림	상대적으로 빠름
패딩 처리	O	X
주요 대상	일반 데이터 전송	오디오 및 비디오 스트리밍
대표 구조	e.g., ECB, CBC	e.g., RC4

# 대칭 암호 원리

- 암호 분석 유형(1/4)
  - 암호문 단독 공격(Ciphertext-Only Attack)
    - 공격자가 암호문만으로 키 혹은 평문을 찾는 공격
      - 암호 알고리즘, 암호문을 알고 있음
      - 전수 공격(Brute-Force Attack)으로 암호문 해독

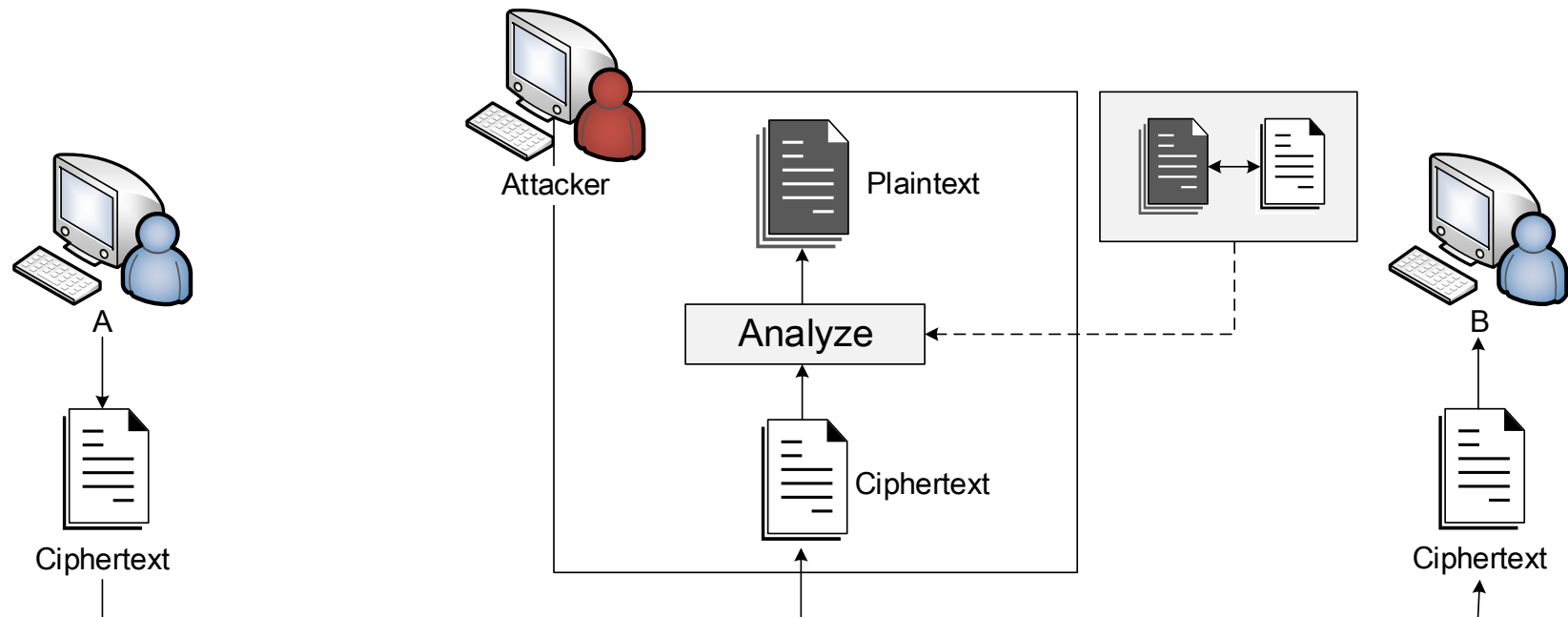


# 대칭 암호 원리

- 암호 분석 유형(2/4)

- 알려진 평문 공격(Known-Plaintext Attack)

- 공격자가 평문 일부와 대응되는 암호문 일부를 이용하여 키 혹은 전체 평문을 찾는 공격
  - 암호 알고리즘, 일부의 평문-암호문 쌍을 알고 있음



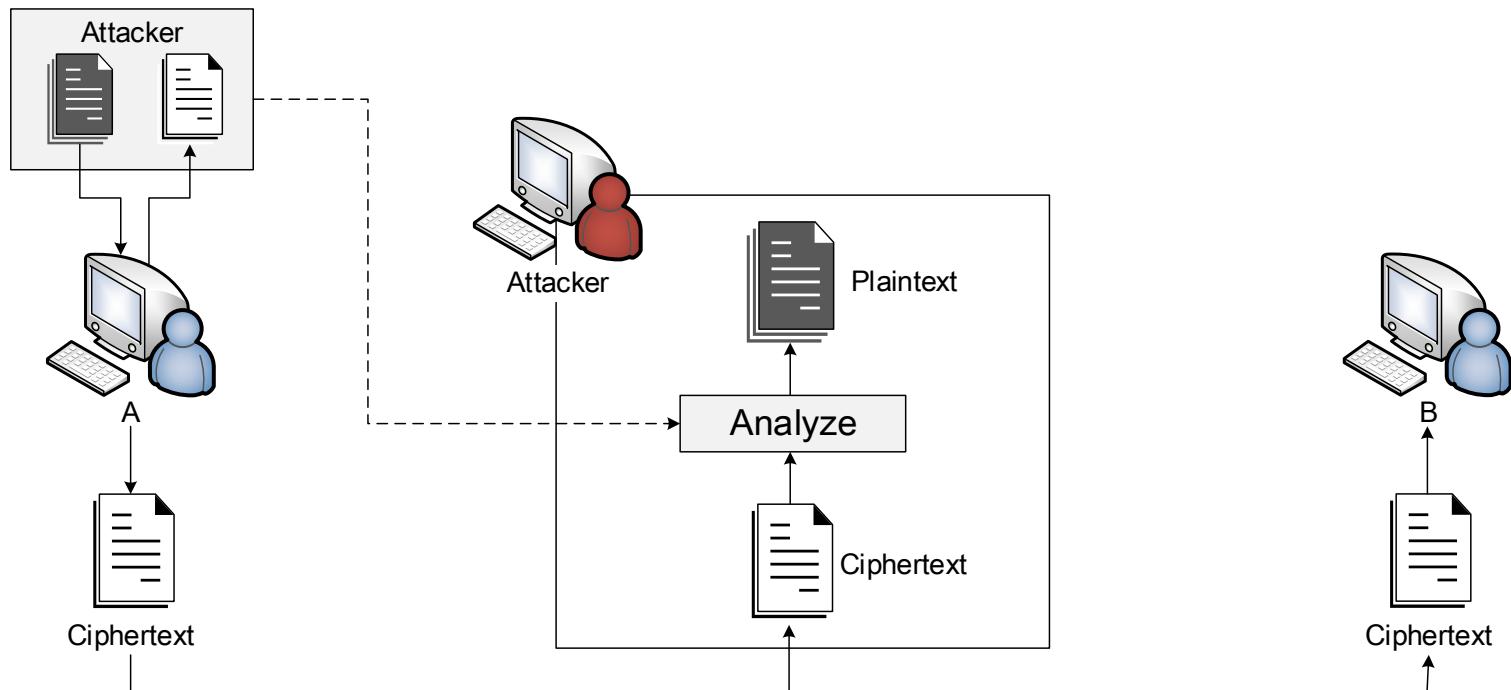
# 대칭 암호 원리

- 암호 분석 유형(3/4)

- 선택 평문 공격(Chosen-Plaintext Attack)

- 공격자가 암호기에 접근하여 평문을 선택하고, 그에 대응하는 암호문을 수집함으로써 키 혹은 평문을 찾는 공격
  - 암호 알고리즘, 암호문을 알고 있음

선택 평문에서  
만들어진 암호문

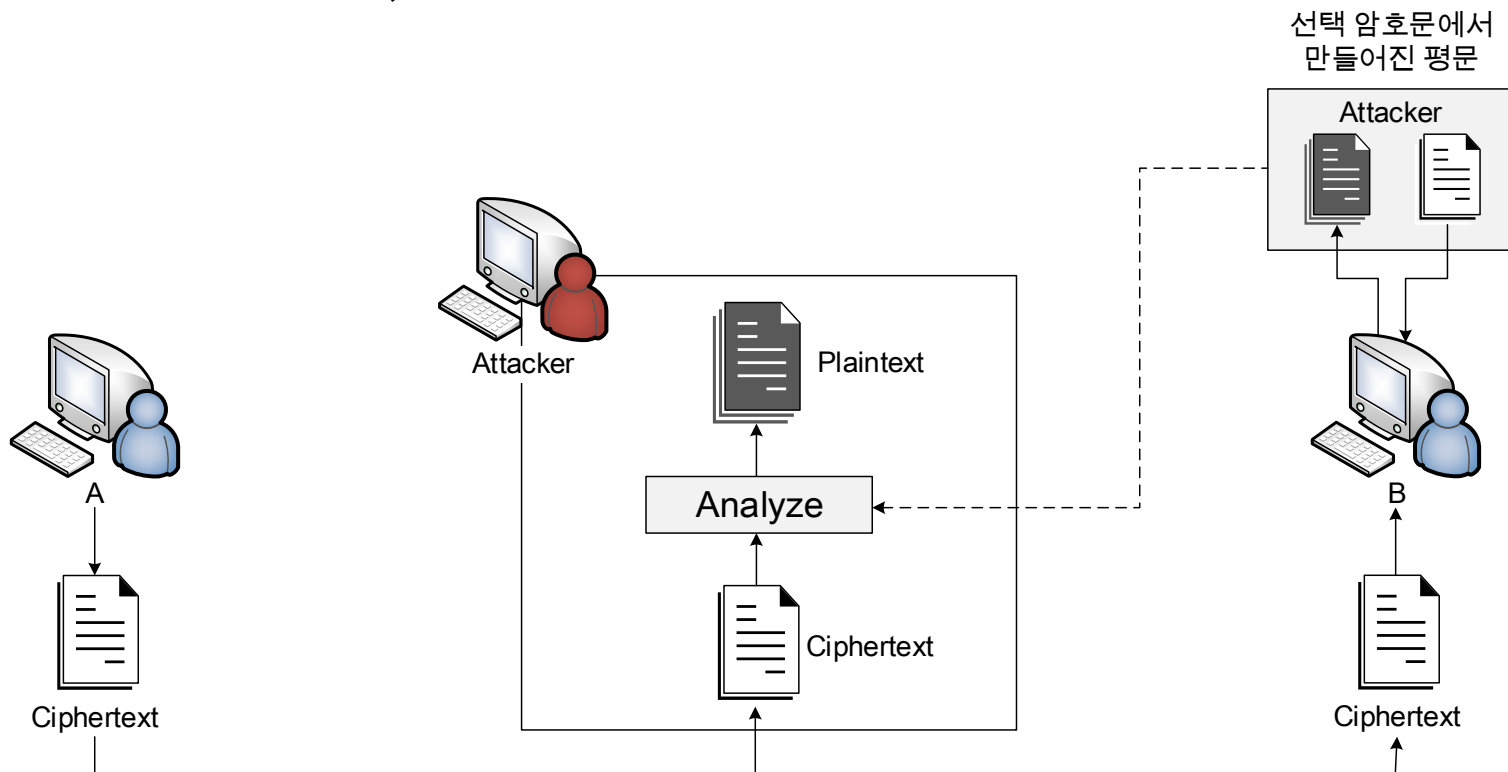


# 대칭 암호 원리

- 암호 분석 유형(4/4)

- 선택 암호문 공격(Chosen-Ciphertext Attack)

- 공격자가 복호기에 접근하여 암호문을 선택하고, 그에 대응하는 평문을 수집함으로써 키를 찾는 공격
  - 암호 알고리즘, 평문을 알고 있음



# 대칭 암호 원리

---

- 대칭 암호(Symmetric Encryption)

- 정의

- 통신자가 동일한 키를 사용하여 데이터를 암호화/복호화하는 암호 방식

- 특징

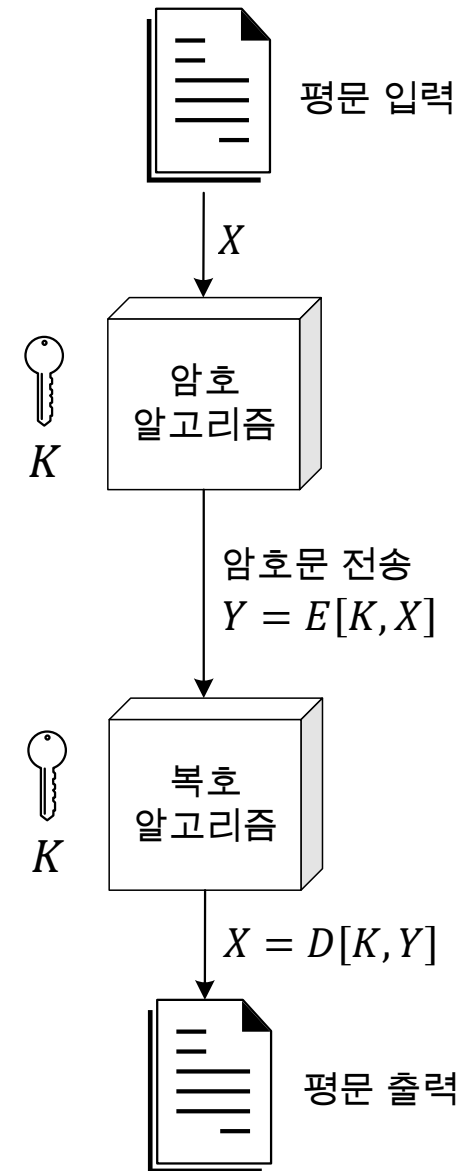
- 암호화 키와 복호화 키가 동일함
- 키를 교환해야 하므로 보안에 취약함
- 키 크기가 작고, 암호 알고리즘 구조가 간단함
- 연산 속도가 빨라 효율적임

# 대칭 암호 원리

- 대칭 암호(Symmetric Encryption)

- 용어

- 평문(Plaintext)
  - 전달할 내용을 담은 일반적인 데이터
- 비밀키(Secret Key)
  - 암호화/복호화를 위해 사용되는 키
- 암호 알고리즘(Encryption Algorithm)
  - 비밀키를 이용하여 평문을 암호화하는 방법
- 암호문(Ciphertext)
  - 암호 알고리즘을 통해 암호화된 평문
- 복호 알고리즘(Decryption Algorithm)
  - 비밀키를 이용하여 암호문을 복호화하는 방법



# 대칭 암호 원리

---

- Feistel 암호

- 정의

- 1973년에 IBM Horst Feistel이 최초로 소개한 대칭 블록 암호 알고리즘

- 특징

- 블록 암호의 대표적인 구조임
  - XOR 연산을 기본 원리로 이용함
  - 암호화/복호화 과정이 동일함
  - 동일한 라운드 함수를 사용함
  - 매라운드마다 다른 키를 사용함

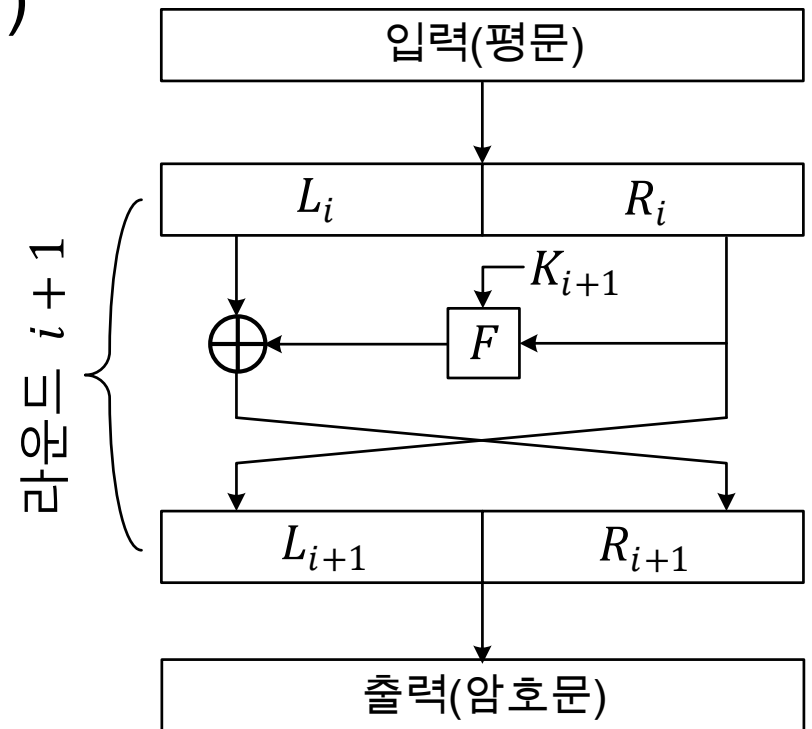


# 대칭 암호 원리

- Feistel 암호

- 암호화 과정

1. 평문 블록 입력
2. 평문 블록을 반으로 나눔( $L_i, R_i$ )
3.  $i$ 라운드 실행( $i = 0 \sim 15$ )
  - $L_{i+1} = R_i$
  - $R_{i+1} = F(R_i, K_{i+1}) \oplus L_i$
4. 16라운드 후,  $L_{16}$ 와  $R_{16}$  교환
5. 암호문 블록 출력

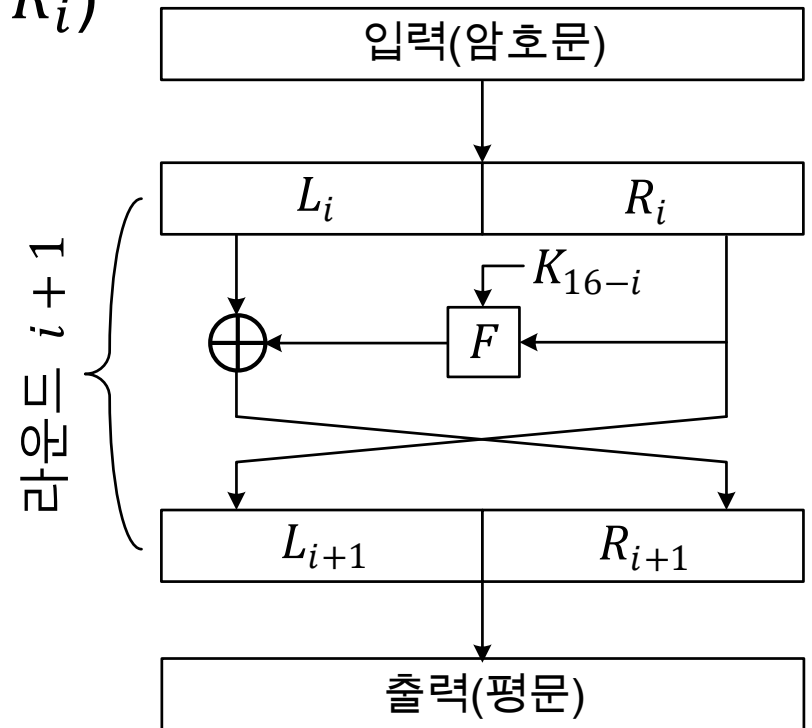


# 대칭 암호 원리

- Feistel 암호

- 복호화 과정

1. 암호문 블록 입력
2. 암호문 블록을 반으로 나눔( $L_i, R_i$ )
3.  $i$ 라운드 실행( $i = 0 \sim 15$ )
  - $L_{i+1} = R_i$
  - $R_{i+1} = F(R_i, K_{16-i}) \oplus L_i$
4. 16라운드 후,  $L_{16}$ 와  $R_{16}$  교환
5. 평문 블록 출력



# 대칭 암호 원리

---

- Feistel 암호
  - 암호화 강도 결정 요소
    - 블록 크기(Block Size)
      - 클수록 강하며, 보통 64비트 사용
    - 키 크기(Key Size)
      - 길수록 강하며, 보통 128비트 사용
    - 라운드 수(Number of Rounds)
      - 여러 번 수행할수록 강하며, 보통 16라운드 수행
    - 서브키 생성 알고리즘(Subkey Generation Algorithm)
      - 복잡할수록 강함
    - 라운드 함수(Round Function)
      - 복잡할수록 강함

# 목 차

---

- 보충
  - 컴퓨터 보안 개념
  - OSI 보안 구조
- 대칭 암호 원리
- 대칭 암호 알고리즘
- 랜덤 넘버와 의사 랜덤 넘버

# 대칭 암호 알고리즘

---

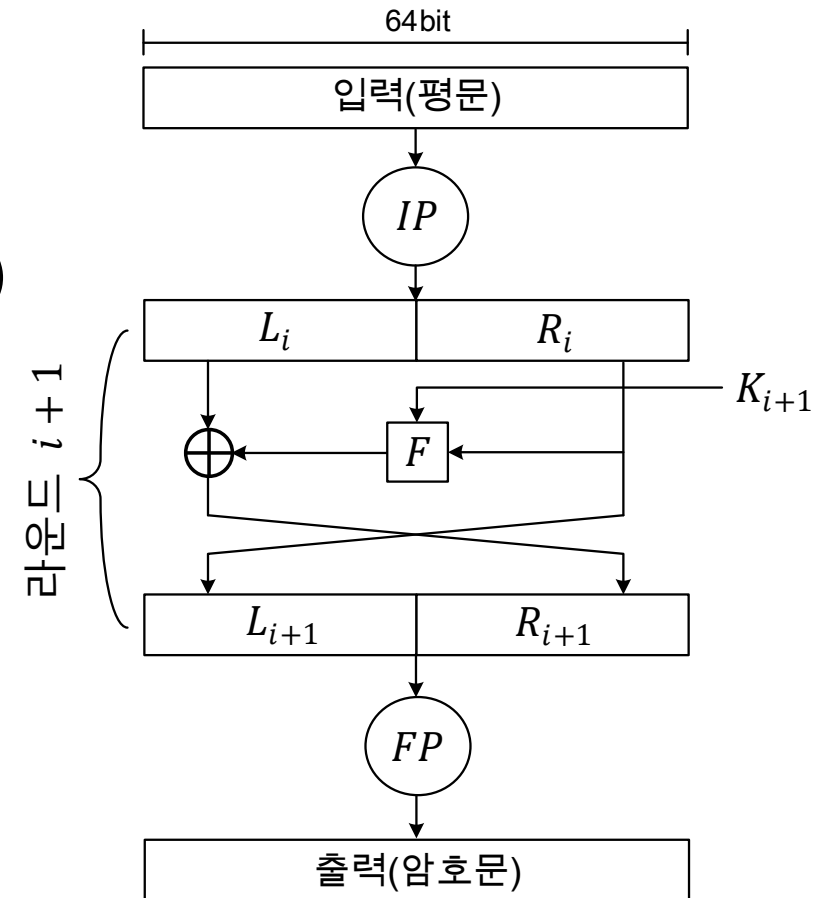
- DES(Data Encryption Standard)
  - 정의
    - 1975년에 NIST(National Institute of Standards and Technology)에서 미국 표준 암호 알고리즘으로 지정한 대칭 블록 암호 알고리즘
  - 특징
    - 64비트의 블록 사용
    - 56비트의 키 사용
    - 16라운드의 Feistel 암호 구조 사용
    - 암호화 과정과 복호화 과정이 동일함
    - 전수 공격(Brute-Force Attack)에 취약함

# 대칭 암호 알고리즘

- DES(Data Encryption Standard)

- 암호화 과정

1. 평문 블록 입력
2.  $IP$ (Initial Permutation)를 거침
3. 평문 블록을 반으로 나눔( $L_i, R_i$ )
4.  $i$ 라운드 실행( $i = 0 \sim 15$ )
  - $L_{i+1} = R_i$
  - $R_{i+1} = F(R_i, K_{i+1}) \oplus L_i$
5. 16라운드 후,  $L_{16}$ 와  $R_{16}$  교환
6.  $FP$ (Final Permutation)를 거침
7. 암호문 블록 출력

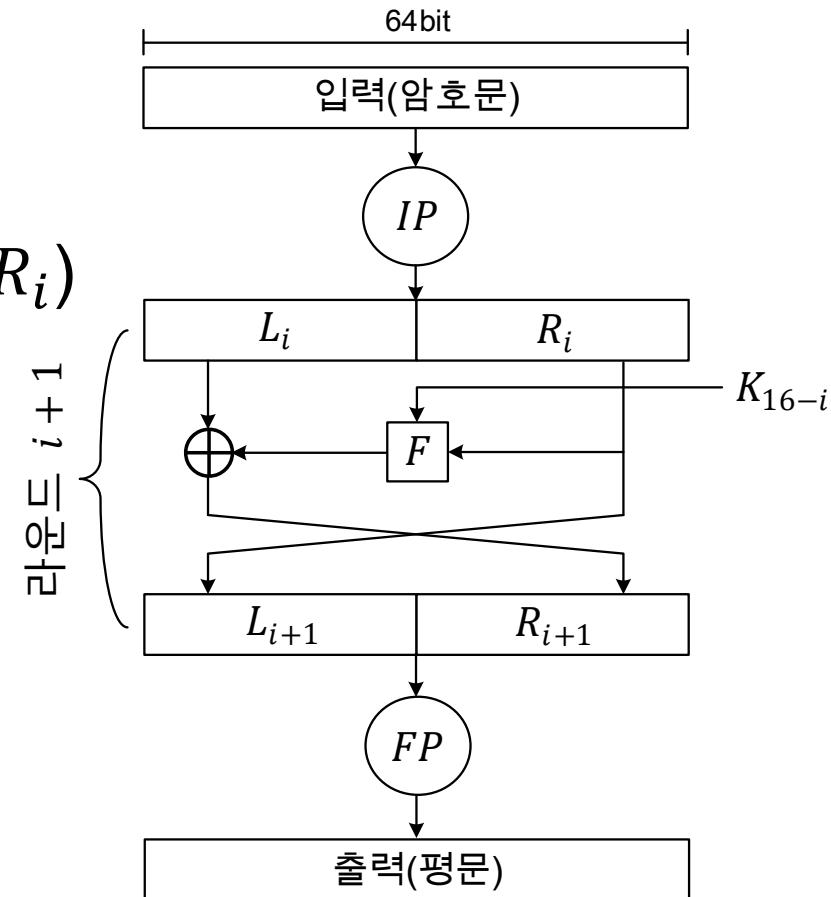


# 대칭 암호 알고리즘

- DES(Data Encryption Standard)

- 복호화 과정

1. 암호문 블록 입력
2.  $IP$ (Initial Permutation)를 거침
3. 암호문 블록을 반으로 나눔( $L_i, R_i$ )
4.  $i$ 라운드 실행( $i = 0 \sim 15$ )
  - $L_{i+1} = R_i$
  - $R_{i+1} = F(R_i, K_{16-i}) \oplus L_i$
5. 16라운드 후,  $L_{16}$ 와  $R_{16}$  교환
6.  $FP$ (Final Permutation)를 거침
7. 평문 블록 출력



# 대칭 암호 알고리즘

- DES(Data Encryption Standard)

- 서브키 생성 과정

1. Parity Drop

- 패리티 비트(8bits) 제거(56bits)

2. 28bits 블록으로 나눔

3. Shift Left

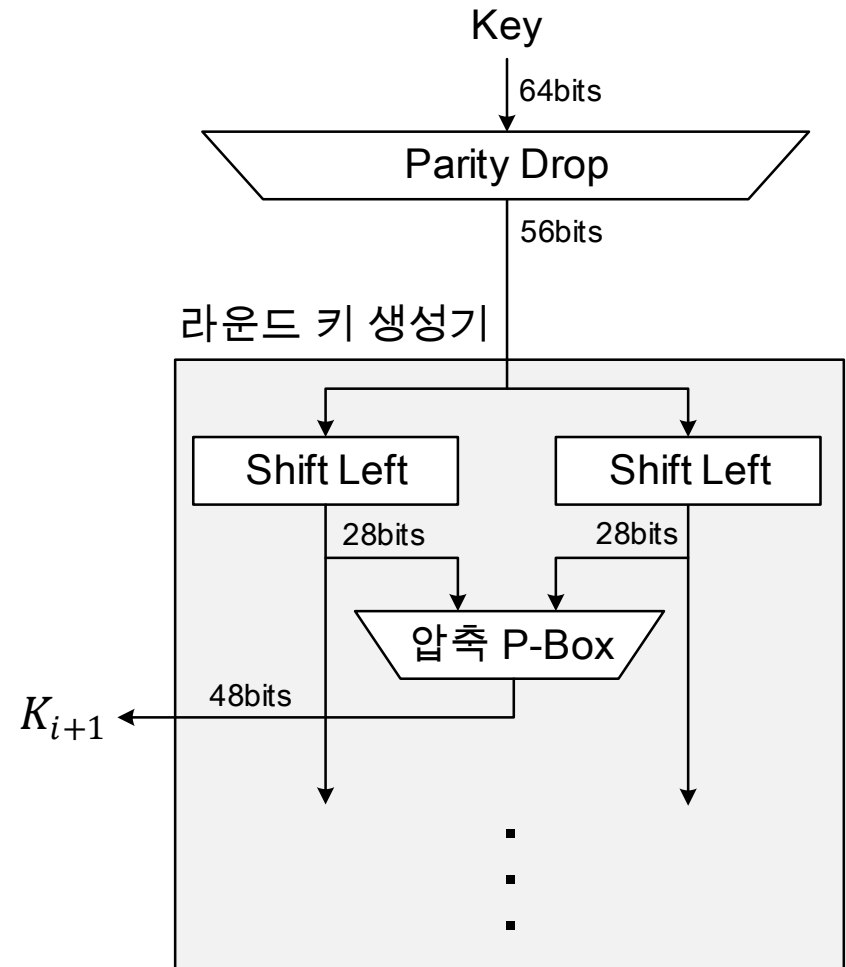
- 1,2,9,16 라운드 키 생성 시,  
1bit 왼쪽 순환 이동
- 나머지 라운드 키 생성 시,  
2bit 왼쪽 순환 이동

4. 압축 P-Box

- 8bits 제거(48bits)

5. 라운드별 서브키( $K_{i+1}$ ) 생성

- 총 16회 반복( $i = 0 \sim 15$ )





# 대칭 암호 알고리즘

- DES(Data Encryption Standard)
  - 초기 치환(*IP*)
    - 초기 치환표를 보고 단순 치환하는 방법
      - e.g., 암호문 1번째 비트 = 평문 58번째 비트
  - 최종 치환(*FP*)
    - 최종 치환표를 보고 단순 치환하는 방법
    - 초기 치환의 역 과정

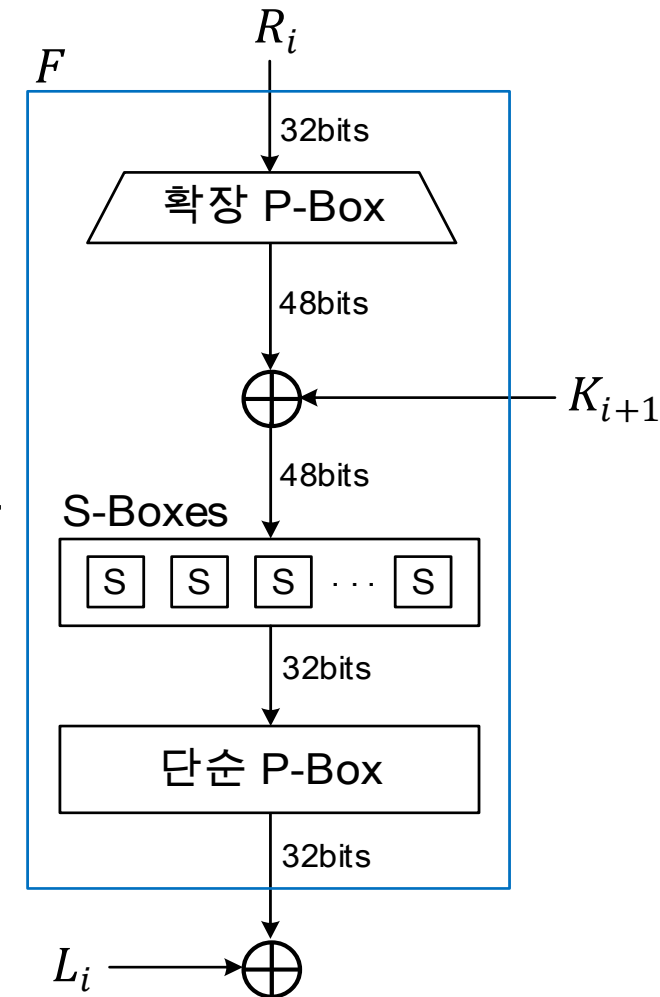
<i>Initial Permutation</i>	<i>Final Permutation</i>
58 50 42 34 26 18 10 02	40 08 48 16 56 24 64 32
60 52 44 36 28 20 12 04	39 07 47 15 55 23 63 31
62 54 46 38 30 22 14 06	38 06 46 14 54 22 62 30
64 56 48 40 32 24 16 08	37 05 45 13 53 21 61 29
57 49 41 33 25 17 09 01	36 04 44 12 52 20 60 28
59 51 43 35 27 19 11 03	35 03 43 11 51 19 59 27
61 53 45 37 29 21 13 05	34 02 42 10 50 18 58 26
63 55 47 39 31 23 15 07	33 01 41 09 49 17 57 25

# 대칭 암호 알고리즘

- DES(Data Encryption Standard)

- 라운드 함수( $F$ )

1. 확장 P-Box로 비트 확장
  - 4bits를 6bits로 확장하는 P-Box 사용
  - $R_i$  (32bits  $\rightarrow$  48bits)
2.  $R_i \oplus K_{i+1}$
3. S-Boxes로 비트 수축
  - 6bits를 4bits로 축소하는 8개의 S-Box 사용
  - $R_i$  (48bits  $\rightarrow$  32bits)
4. 단순 P-Box로 함수값 반환



# 대칭 암호 알고리즘

- DES(Data Encryption Standard)

- P-Box(Permutation-Box)

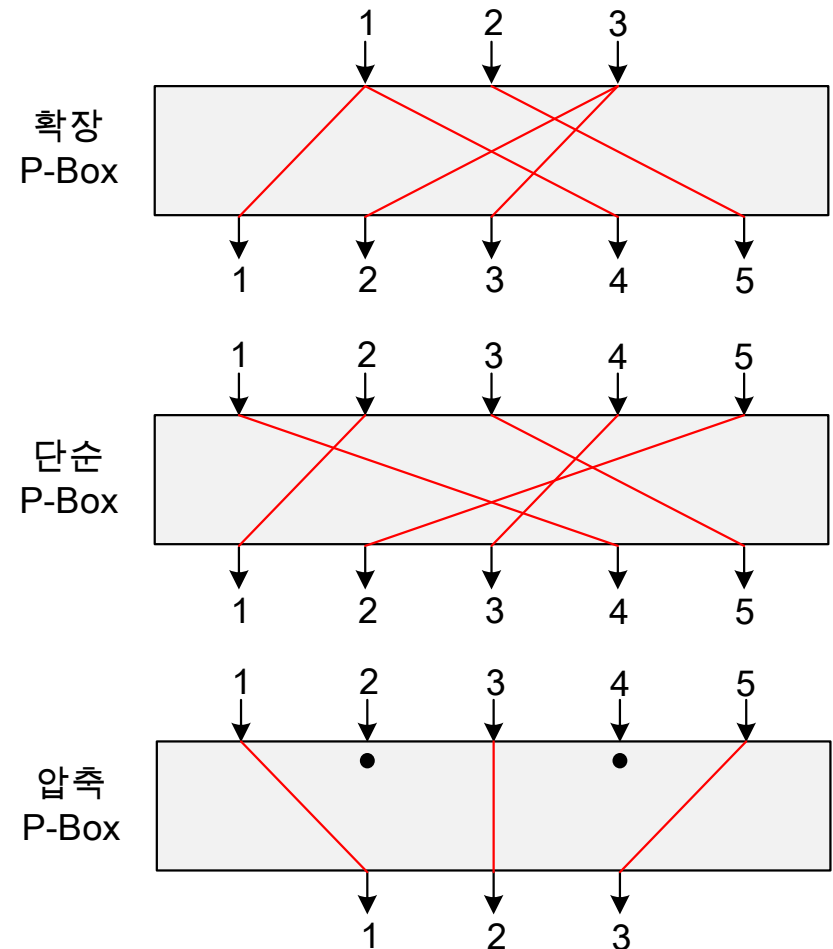
- 비트의 위치를 변환하는 방법
- 확장(Expansion) P-Box
  - N bits를 M bits로 확대 및 변환
    - e.g., DOG -> DGGDO

- 단순(Straight) P-Box

- N bits를 N bits로 변환
  - e.g., HELLO -> EOLHL

- 압축(Compression) P-Box

- N bits를 M bits로 축소 및 변환
  - e.g., HELLO -> HLO



# 대칭 암호 알고리즘

- DES(Data Encryption Standard)
  - S-Box(Substitution-Box)
    - 비트를 수학적 규칙에 의해 치환하는 방법
    - 설계자의 S-Box Table을 활용함
      - e.g., 6bits(011011)의 입력 값을 4bits로 축소시키는 방법
        - 6bits 중 1과 6은 행, 2~5는 열을 결정
        - 행: 01 => 1, 열: 1101 => 13
        - 011011(6bits) -> 1001(4bits)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

# 대칭 암호 알고리즘

---

- 3DES(Triple Data Encryption Standard)

- 정의

- DES 알고리즘을 세 번 수행하는 대칭 블록 암호 알고리즘

- 특징

- 64비트의 블록으로 3번의 DES 알고리즘 수행
    - 암호화 과정: 암호화-복호화-암호화
    - 복호화 과정: 복호화-암호화-복호화
  - 2, 3개의 키와 48라운드의 Feistel 암호 구조 사용
    - 2개인 경우, 112비트의 키
    - 3개인 경우, 168비트의 키
  - DES의 한계를 보완하는 알고리즘임
  - DES보다 암호화/복호화 속도가 느림

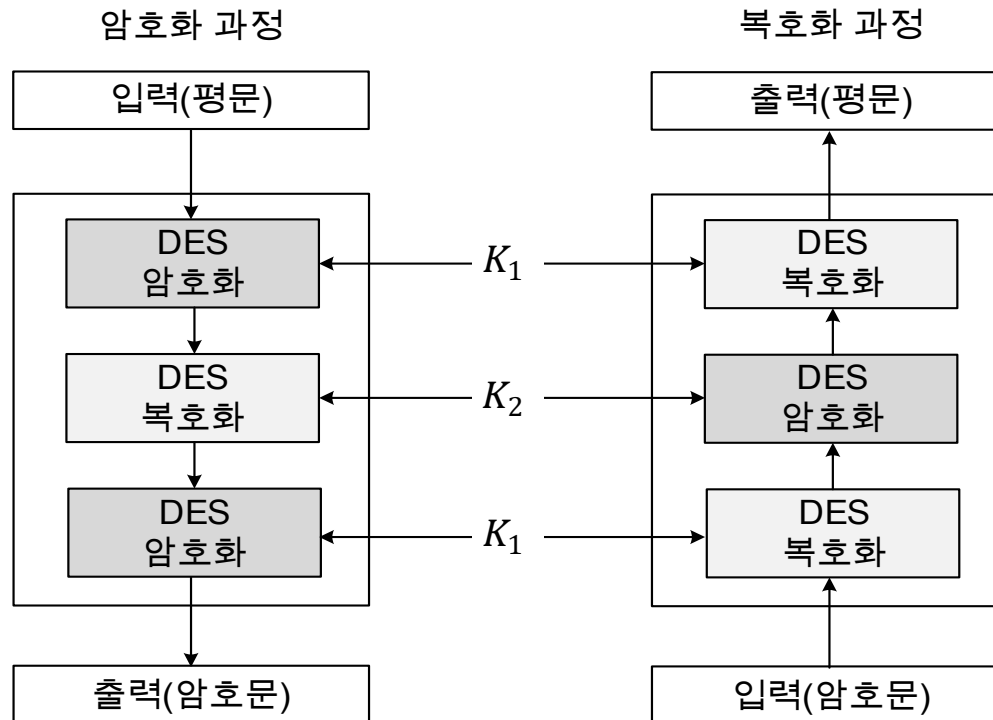
# 대칭 암호 알고리즘

- 3DES(Triple Data Encryption Standard)

- 2개의 키를 사용한 암호화/복호화 과정

- 암호화:  $C = E(K_1, D(K_2, E(K_1, P)))$

- 복호화:  $P = D(K_1, E(K_2, D(K_1, C)))$



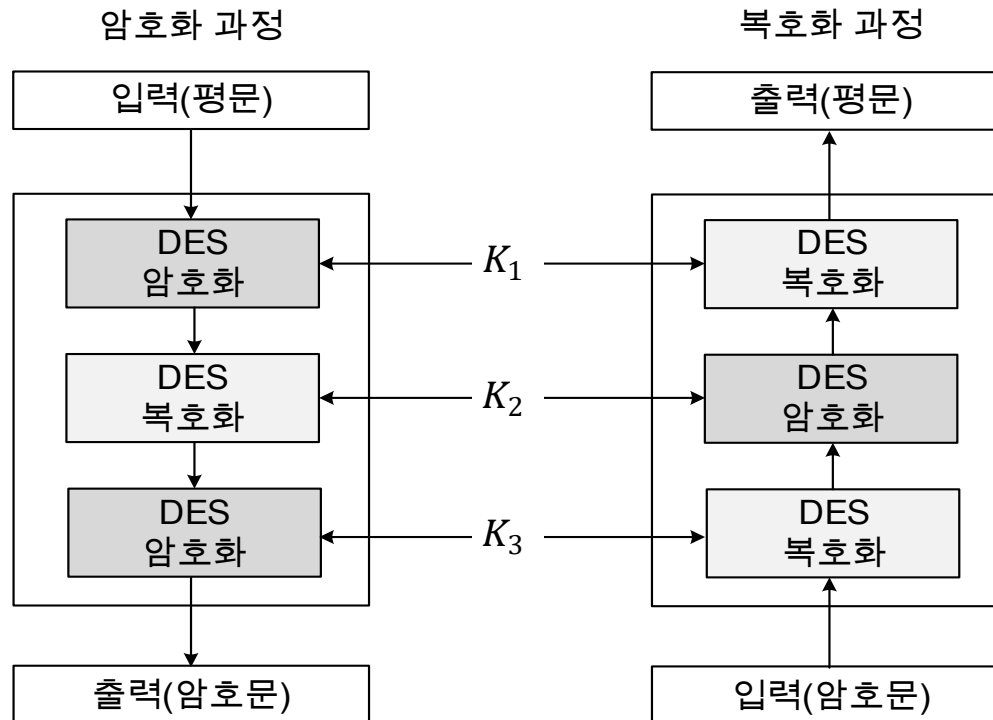
# 대칭 암호 알고리즘

- 3DES(Triple Data Encryption Standard)

- 3개의 키를 사용한 암호화/복호화 과정

- 암호화:  $C = E(K_3, D(K_2, E(K_1, P)))$

- 복호화:  $P = D(K_1, E(K_2, D(K_3, C)))$



# 대칭 암호 알고리즘

---

- AES(Advanced Encryption Standard)
  - 정의
    - 2001년에 NIST에 의해 제정된 대칭 블록 암호 알고리즘
  - 특징
    - 128비트의 블록 사용
    - 128, 192, 256비트의 키 사용
      - 각각 10, 12, 14라운드 사용
    - SPN(Substitution-Permutation Network) 암호 구조 사용
      - Feistel 암호 구조와 달리, 역함수 필요
    - DES와 3DES의 한계를 보완하는 알고리즘임

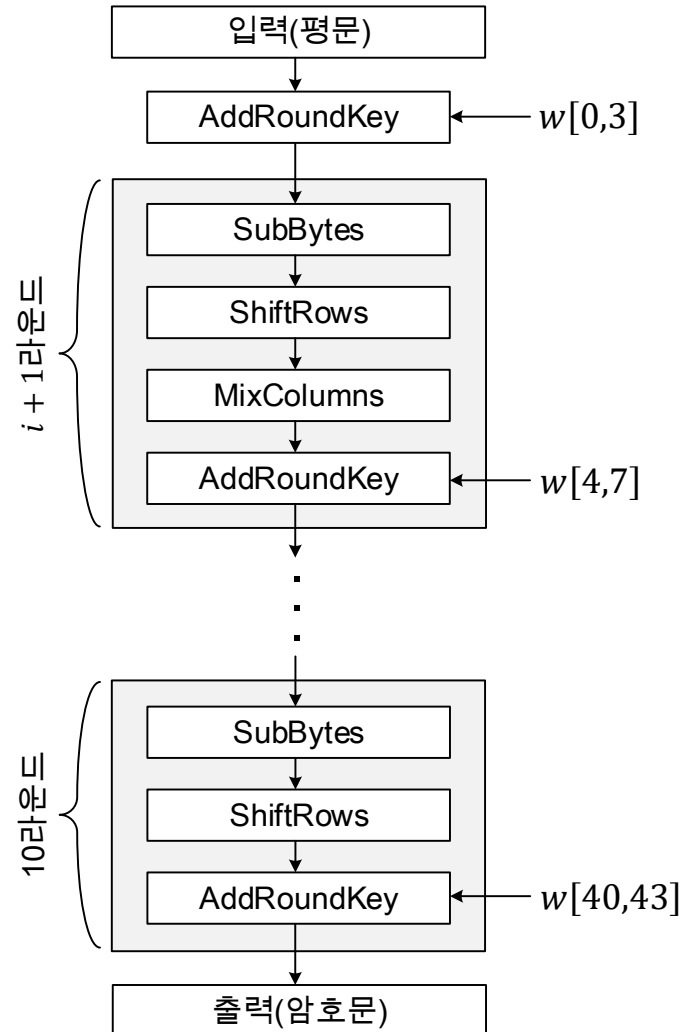


# 대칭 암호 알고리즘

- AES(Advanced Encryption Standard)

- 암호화 과정

1. 평문 블록 입력
2. AddRoundKey 실행
3.  $i$ 라운드 실행( $i = 0 \sim 8$ )
  - SubBytes
  - ShiftRows
  - MixColumns
  - AddRoundKey
4. 10라운드 실행
  - SubBytes
  - ShiftRows
  - AddRoundKey
5. 암호문 블록 출력

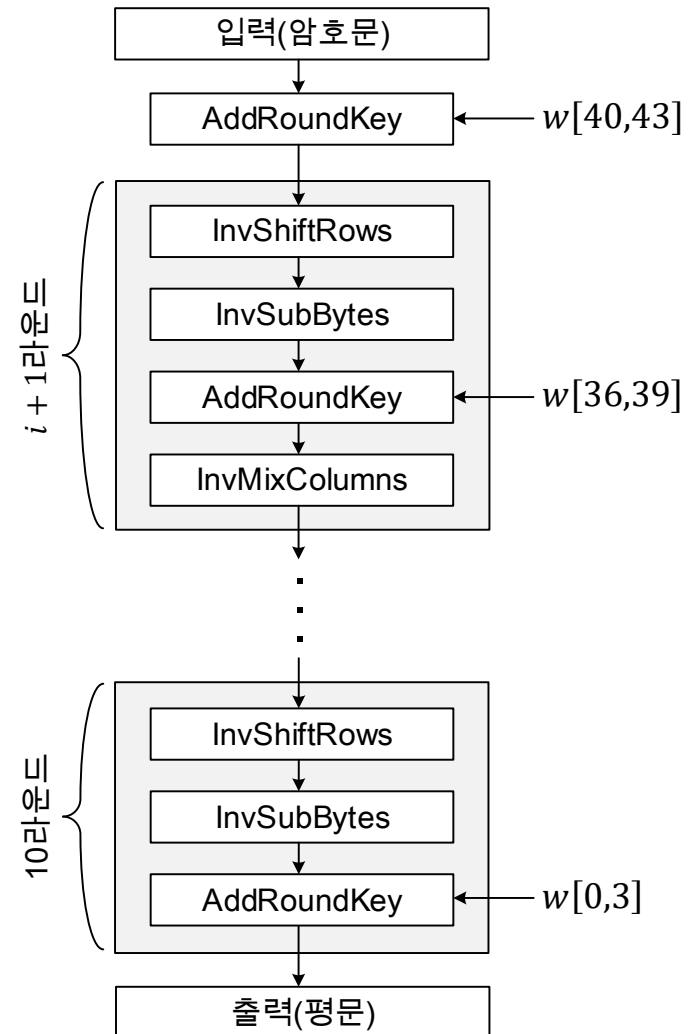


# 대칭 암호 알고리즘

- AES(Advanced Encryption Standard)

- 복호화 과정

1. 암호문 블록 입력
2. AddRoundKey 실행
3.  $i$ 라운드 실행( $i = 0 \sim 8$ )
  - InvShiftRows
  - InvSubBytes
  - AddRoundKey
  - InvMixColumns
4. 10라운드 실행
  - InvShiftRows
  - InvSubBytes
  - AddRoundKey
5. 평문 블록 출력



# 대칭 암호 알고리즘

## • AES(Advanced Encryption Standard)

### • Round Key Schedule

#### 1. 이전 라운드의 마지막 열 변환

- 한 칸씩 위로 Shift
- SubBytes 적용

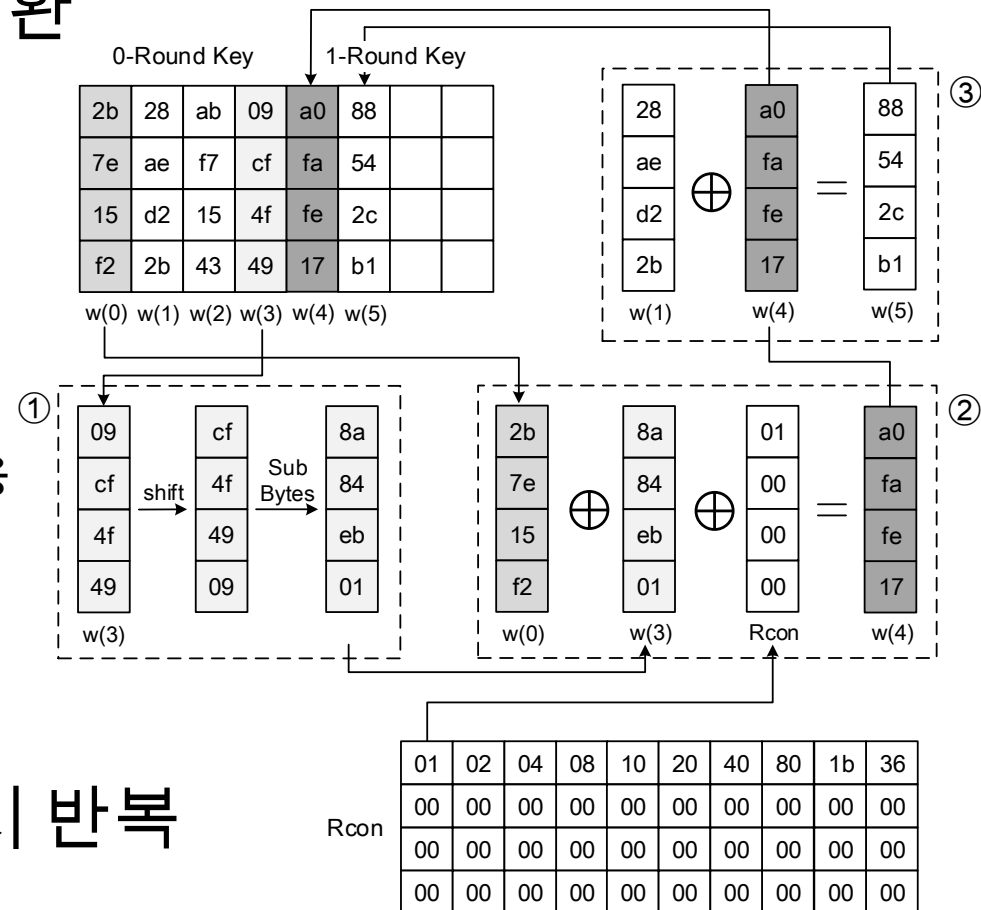
#### 2. 라운드의 첫열 계산

- 이전 라운드 첫열과 마지막 열, Rcon의 한 열 XOR 연산
  - Rcon은 라운드마다 한 열씩 사용

#### 3. 라운드의 나머지 열 계산

- 이전 라운드 나머지 열과 해당 라운드 열 XOR 연산

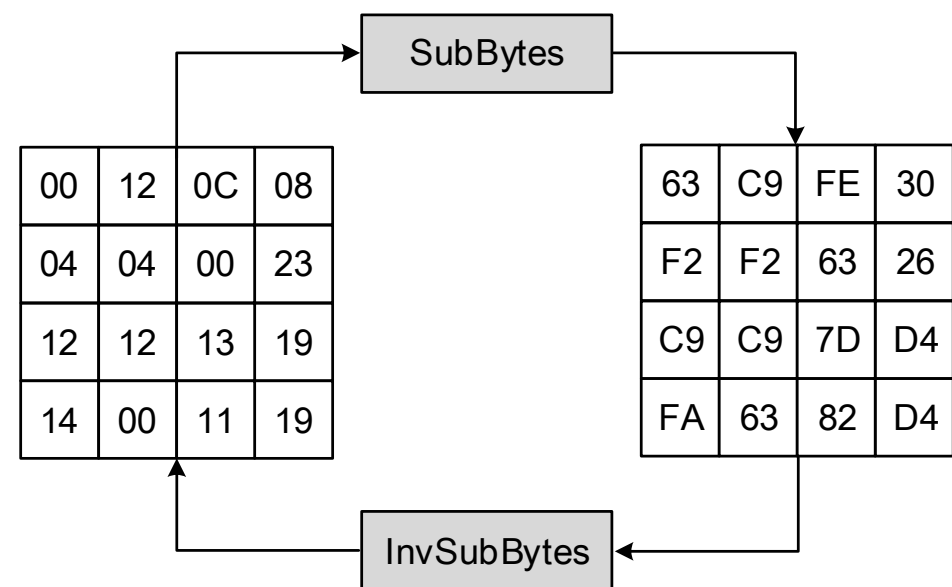
#### 4. 1~3의 과정을 10라운드까지 반복



# 대칭 암호 알고리즘

- AES(Advanced Encryption Standard)
  - SubBytes(Substitution Bytes)
    - Substitution Table을 통해 각 비트 블록을 바이트로 대체
  - InvSubBytes
    - SubBytes와 동일하나, 역 Table은 따로 존재함

		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	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

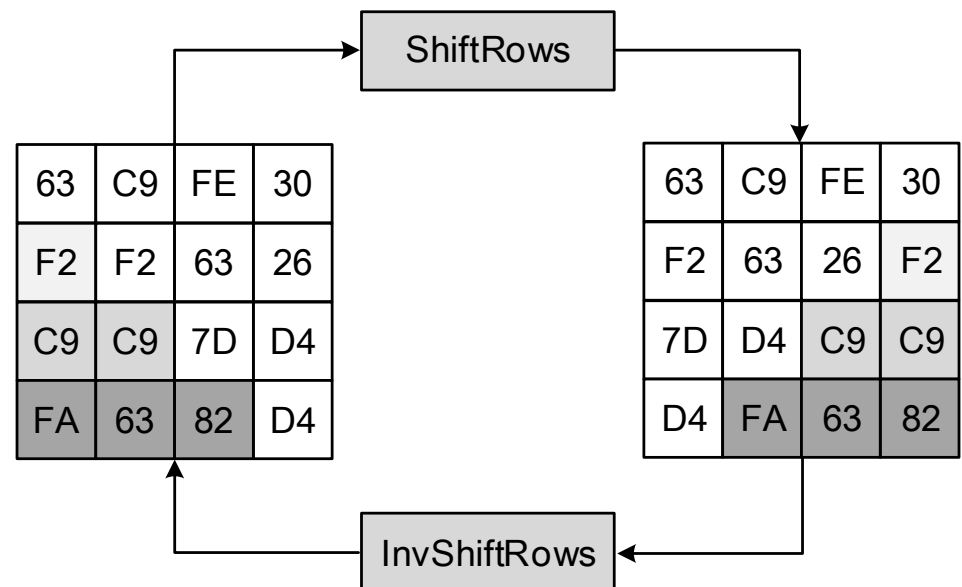
- AES(Advanced Encryption Standard)

- ShiftRows

- 첫 번째 위치부터 각 행의 위치가 증가되는 수만큼 행렬의 각 행을 왼쪽으로 순환 이동
  - 첫 번째 행은 이동되지 않음
  - 두 번째 행은 한 자리씩 이동
  - 세 번째 행은 두 자리씩 이동
  - 네 번째 행은 세 자리씩 이동

- InvShiftRows

- ShiftRows와 동일하나, 오른쪽으로 순환 이동



# 대칭 암호 알고리즘

- AES(Advanced Encryption Standard)

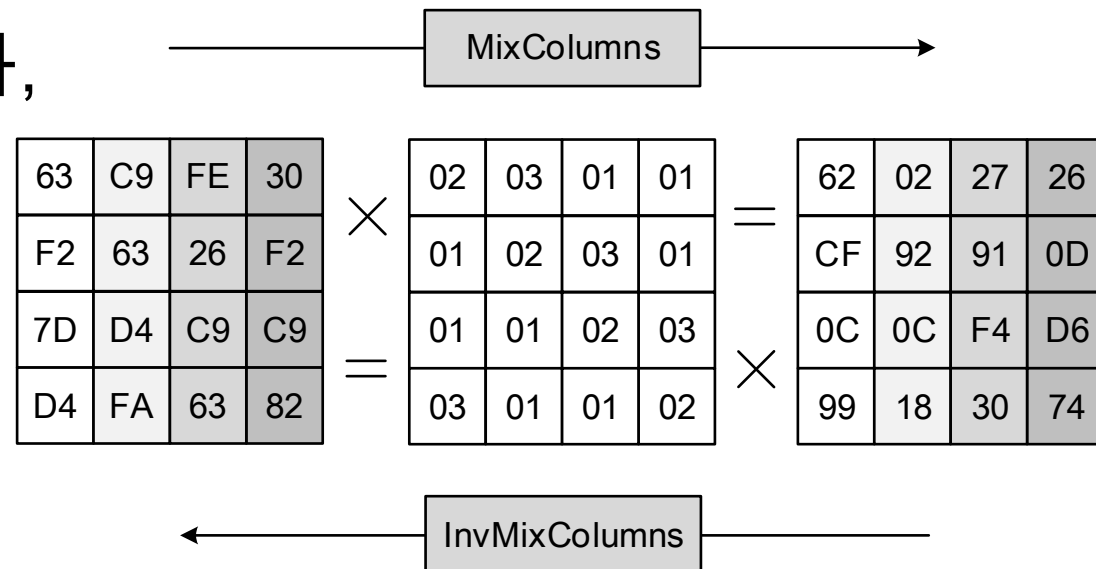
- MixColumns

- 특정 행렬과 기존 상태를 행렬 곱셈을 이용하여 열 단위로 섞음

$$\begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix} \times \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} = \begin{bmatrix} \phantom{00} \\ \phantom{00} \\ \phantom{00} \\ \phantom{00} \end{bmatrix} \begin{array}{l} \leftarrow ax + by + cz + dt \\ \leftarrow ex + fy + gz + ht \\ \leftarrow ix + jy + kz + lt \\ \leftarrow mx + ny + oz + pt \end{array}$$

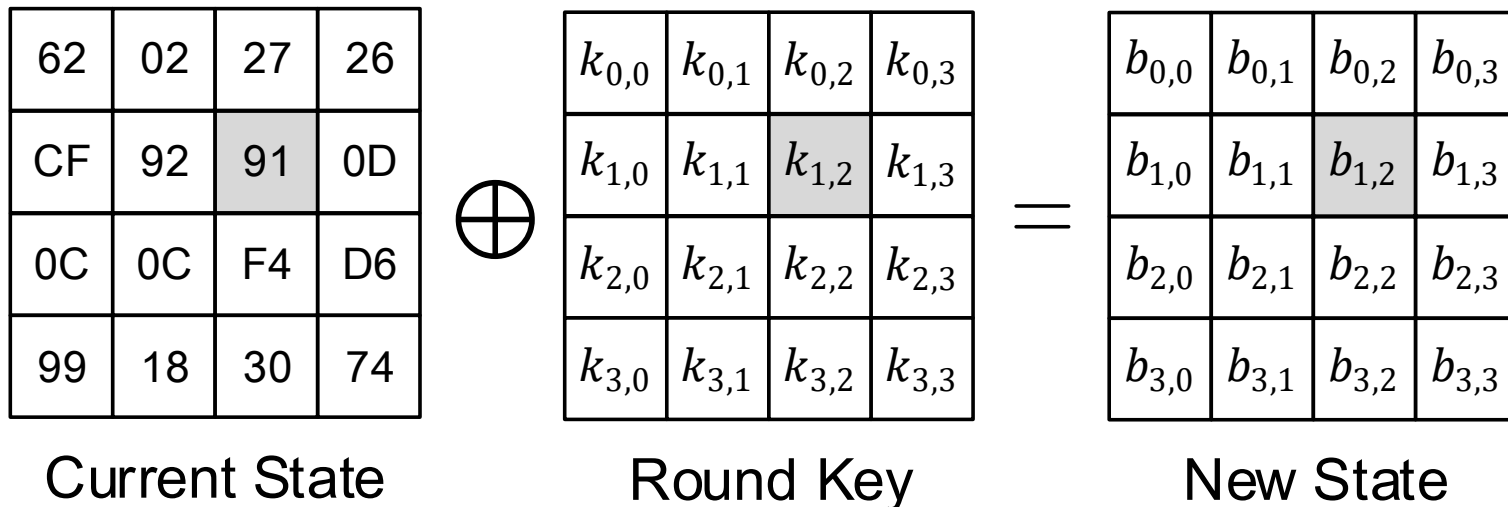
- InvMixColumns

- MixColumns와 동일하나, 암호화된 상태를 특정 행렬과 곱함
  - 특정 역행렬이 존재함



# 대칭 암호 알고리즘

- AES(Advanced Encryption Standard)
  - AddRoundKey
    - 현재 상태와 라운드 키를 비트 단위로 XOR 연산
      - 키 스케줄링에 따라 생성되는 라운드 키는 라운드마다 다름



# 대칭 암호 알고리즘

## • 암호 알고리즘 비교

특징	DES	3DES	AES
암호 구조	Feistel	Feistel	SPN
블록 크기(bit)	64	64	128
키 길이(bit)	56	112/168	128/192/256
라운드 수	16	48	10/12/14
상대적인 속도	보통	느림	빠름



# 목 차

---

- 보충
  - 컴퓨터 보안 개념
  - OSI 보안 구조
- 대칭 암호 원리
- 대칭 암호 알고리즘
- 랜덤 넘버와 의사 랜덤 넘버

# 랜덤 넘버와 의사 랜덤 넘버

---

- 랜덤 넘버(Random Number)

- 정의

- 특정 배열 순서나 규칙을 가지지 않는 연속적인 임의의 수

- 특징

- 무작위성(Randomness)

- 통계적 편중 없이 수열이 무작위로 되어 있다는 성질

- 균등분포(Uniform Distribution)

- 수열의 비트 분포가 반드시 균등해야 함

- 독립성(Independence)

- 수열에서 추출한 부분수열은 다른 수열로부터 추론 불가능해야 함

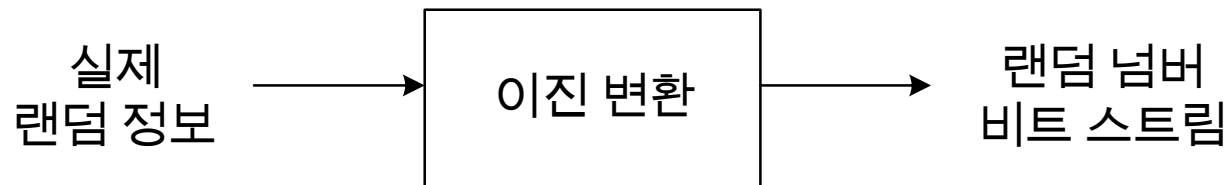
- 예측 불가능성(Unpredictability)

- 과거의 수열로부터 다음 수를 예측할 수 없다는 성질

# 랜덤 넘버와 의사 랜덤 넘버

---

- 랜덤 넘버(Random Number)
- 랜덤 넘버 생성기
  - TRNG(True Random Number Generator)
    - 실제로 랜덤한 정보를 입력으로 사용하여 랜덤 넘버 생성
      - e.g., 키보드 입력 패턴, 마우스 움직임
    - 해당 입력을 조합하여 랜덤 바이너리 출력



# 랜덤 넘버와 의사 랜덤 넘버

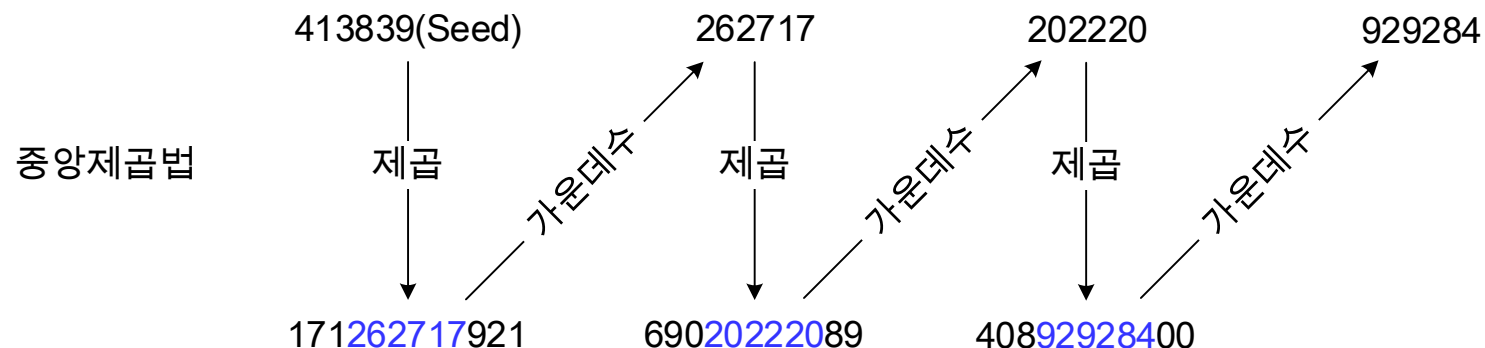
- 의사 랜덤 넘버(Pseudo Random Number)

- 정의

- 초기값을 이용하여 컴퓨터 알고리즘으로 만드는 랜덤 넘버

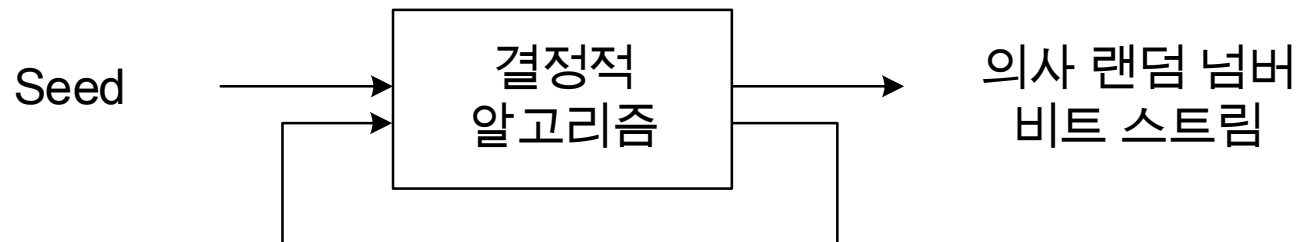
- 특징

- 컴퓨터 알고리즘과 같은 소프트웨어로 구현됨
  - e.g., 노이만의 중앙제곱법
- TRNG로 생성한 진성 랜덤 넘버를 Seed로 사용함
  - Seed는 내부 상태 초기화를 위해 사용



# 랜덤 넘버와 의사 랜덤 넘버

- 의사 랜덤 넘버(Pseudo Random Number)
- 의사 랜덤 넘버 생성기
  - PRNG(Pseudo Random Number Generator)
    - 고정값 Seed를 입력받아 알고리즘을 이용해 출력 비트열 생성
    - 무한한 의사 랜덤 넘버 비트열 생성 시 사용
  - PRF(Pseudo Random Function)
    - 고정값 Seed를 입력받아 알고리즘을 이용해 출력 비트열 생성
    - 고정된 길이의 의사 랜덤 넘버 비트열 생성 시 사용



---

# Thanks!

김 지 혜 ([jihye@pel.sejong.ac.kr](mailto:jihye@pel.sejong.ac.kr))