2023/01/11, 2023 겨울방학 보안기초 세미나

대칭 암호와 메시지 기밀성

- 2.1~2.3 -

강 민 채(minchae@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

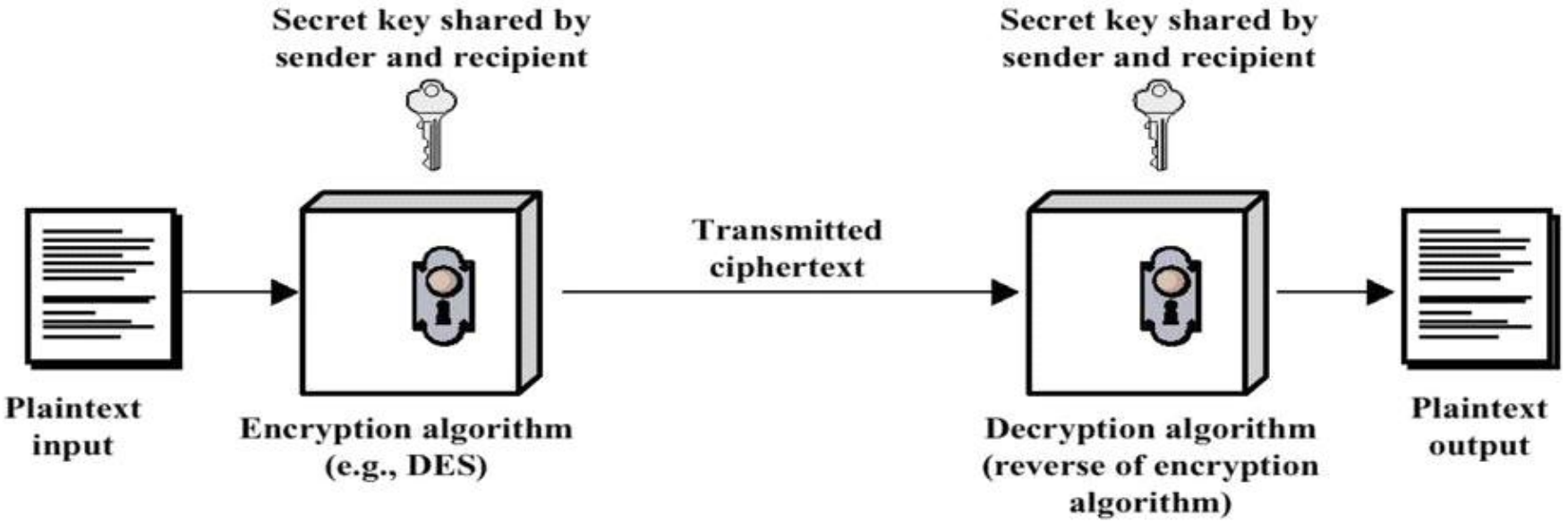
대칭 암호

- 대칭 암호(관용 암호, 비밀 키 암호, 단일 키 암호)
- 공개 키 개발 이전 사용되던 유일한 암호
- 아직도 가장 많이 사용되고 있는 암호

대칭 암호의 5가지 요소

- **평문**
  - 알고리즘의 입력으로 이용되는 원문
- **암호 알고리즘**
  - 입력으로 들어온 원문을 치환, 변환하는 절차
- **비밀 키**
  - 알고리즘의 입력으로 이용
  - 보안에서 중요
- **암호문**
  - 평문과 비밀키에 따라 달라짐
- **복호 알고리즘**
  - 암호 알고리즘을 역수행하여 원문을 복구하는 절차

대칭 암호 모델


Secret key shared by sender and recipient
Secret key shared by sender and recipient
Transmitted ciphertext
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

대칭 키 암호를 안전하게 사용하기 위한 조건

- **강한 암호 알고리즘이 있어야 한다**

  - 암호문과 알고리즘을 알고 있어도 위험이 없어야 함

- **송수신자는 공유하는 비밀 키를 안전하게 획득, 보관해야한다**

  - 비밀 키가 노출되면 그 키를 이용한 모든 통신은 노출되기 때문

-> 즉, 대칭 암호의 보안은 키의 비밀성에 의해 지켜진다

암호 시스템의 3가지 단계

- **평문을 암호문으로 전환하는데 사용되는 연산 유형**
  - 대체
  - 치환
- **사용되는 키의 수**
  - 송수신자 양측이 동일한 키를 사용: 대칭 암호
  - 송수신자 양측이 다른 키를 사용: 비대칭 암호
- **평문 처리 방법**
  - 블록 암호
  - 스트림 암호

암호 해독 공격 유형

공격 유형	암호 해독가가 알고 있는 정보
암호문만 알고 있는 공격 * 전수 공격 이용 가능	- 암호 알고리즘 - 해독해야 할 암호문
알려진 평문 공격 * 예측되는 단어 공격	- 암호 알고리즘 - 해독해야 할 암호문 - 비밀 키로 만들어진 한 쌍 혹은 여러 쌍의 평문-암호문
선택 평문 공격	- 암호 알고리즘 - 해독해야 할 암호문 - 해독가가 선택한 평문 메시지와 비밀 키로 그 평문을 암호화 한 암호문
선택 암호문 공격	- 암호 알고리즘 - 해독해야 할 암호문 - 해독가가 목적을 갖고 선택한 암호문과 비밀 키로 그 암호문을 복호화한 평문
선택문 공격	- 암호 알고리즘 - 해독해야 할 암호문 - 해독가가 선택한 평문 메시지와 비밀 키로 그 평문을 암호화 한 암호문 - 해독가가 목적을 갖고 선택한 암호문과 비밀 키로 그 암호문을 복호화한 평문

암호 구조의 안전 조건

- **암호문을 깨는데 드는 비용이 암호화된 정보의 가치보다 크다**
- **암호문을 깨는데 걸리는 시간이 해당 정보의 수명보다 길다**

-> 암호를 깨는데 드는 노력을 정량화하는게 어렵지만,
전수 공격 방법으로 가능

페이스텔 암호 구조

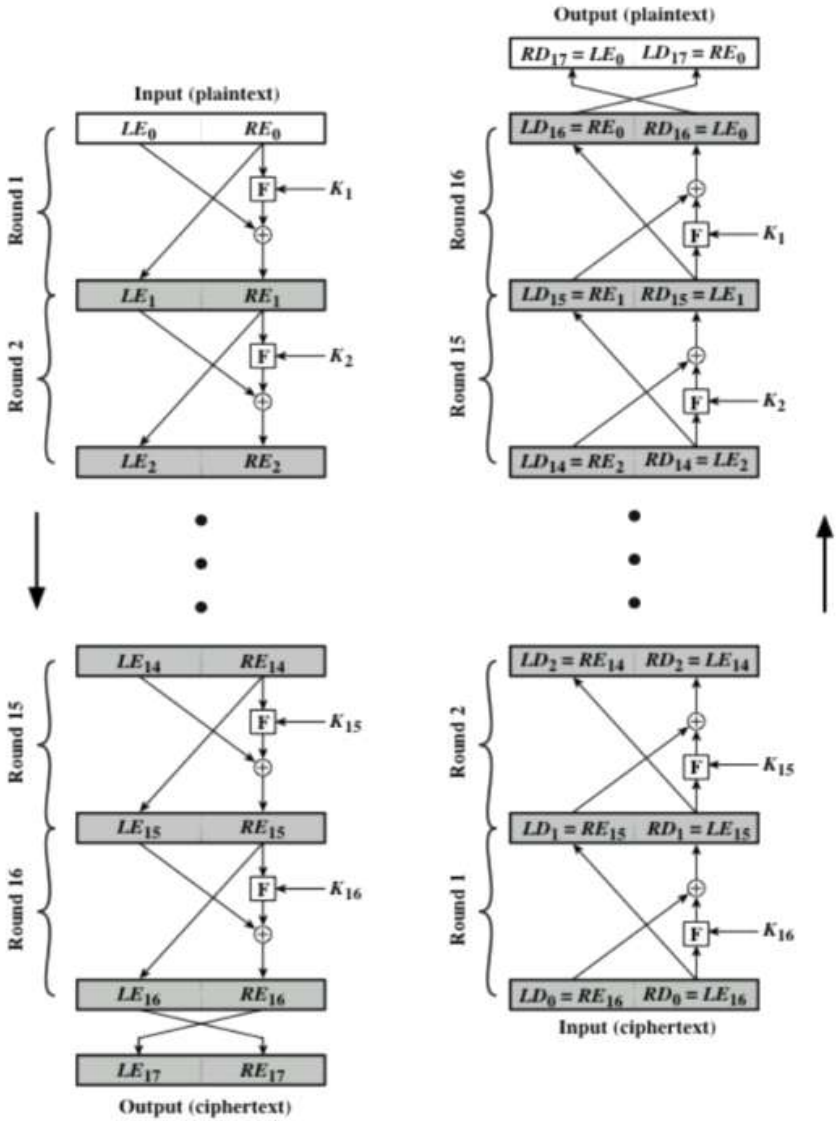


Figure 2.2 Feistel Encryption and Decryption (16 rounds)

암호화:
1) 평문 블록을 두 조각으로 나눈다
2) 대체: 왼쪽 반의 데이터를 오른쪽 반의 데이터에 라운드 함수를 적용한 것과 XOR한것으로 대체한다
* 라운드 함수를 이용시, 매 과정 다른 라운드 서브키를 사용한다
3) 치환: 두 개의 반쪽짜리 데이터를 치환한다
-> 이 과정의 라운드를 16라운드까지 수행

복호화:
근본적으로 암호 과정과 동일하지만,
서브키의 순서를 거꾸로 적용한다

-> 암호 알고리즘과 복호 알고리즘이 같은 알고리즘이다

페이스텔 암호에서의 매개변수와 설계특성

- **블록 길이**
  - 길수록 보안이 강하지만, 암호화/복호화 속도가 떨어짐
  - 64비트 블록 길이가 합리적임
- **키 길이**
  - 길수록 보안이 강하지만, 암호화/복호화 속도가 떨어짐
  - 보편적인 키 길이는 128비트
- **라운드 수**
  - 라운드 수가 증가할수록 보안이 강함
  - 전형적인 라운드 수는 16
- **서브키 생성 알고리즘**
  - 복잡할수록 해독이 어려움
- **라운드 함수**
  - 복잡할수록 해독이 어려움

대칭 블록 암호 설계시 고려 사항

- **빠른 소프트웨어 암호/복호**

  - 하드웨어적인 구현이 아닌, 응용 프로그램이나 유틸리티 함수에 암호화를 내장하여 알고리즘의 실행 속도를 고려

- **용이한 해독**

  - 알고리즘 구조를 단순하게 만들어 암호 해독적 취약점을 찾기 쉽게 만듬

3가지 블록 암호 알고리즘

- **DES(Data Encryption Standard)**
- **3중 DES(3DES)**
- **AES(Advanced Encryption Standard)**

DES(Data Encryption Standard)

- **평문 길이: 64비트**
- **키 길이: 56비트**
- **16개 서브키 사용**
- **페이스텔 네트워크의 변형된 형태**
- **복호 과정이 암호 과정과 동일하다**

DES의 강도

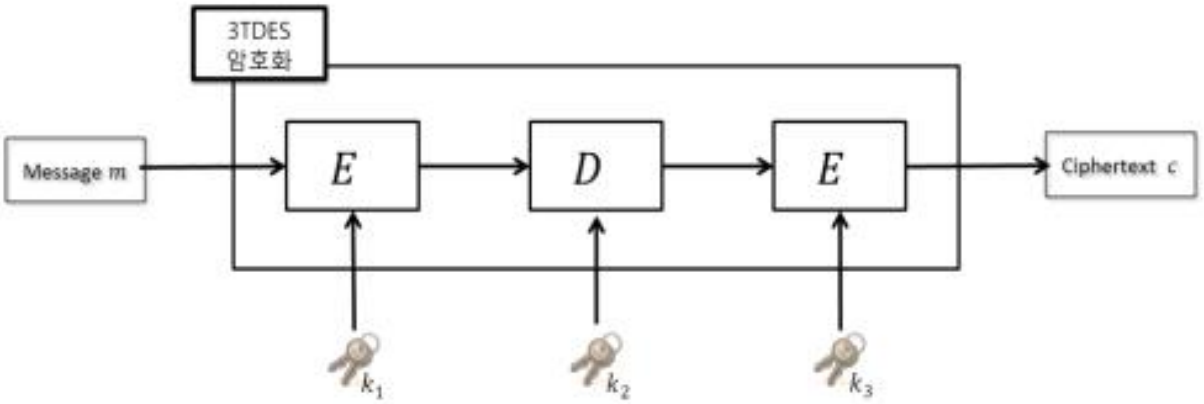
- DES는 수많은 시도에도 취약점이 발견되지 못함
- 하지만, 키의 길이가 56비트

-> 전수공격 시 다중코어 컴퓨터(10^9복호화/초 속도)로는 1년, 슈퍼 컴퓨터(10^{13}복호화/초 속도)로는 1시간만에 비밀키를 찾을 수 있음

즉, 다른 암호 알고리즘 필요

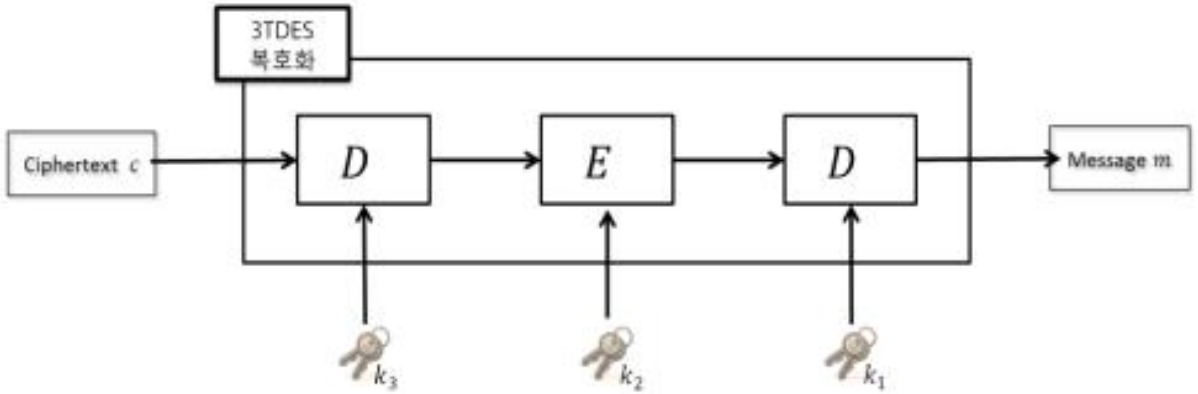
3중 DES

- DES 알고리즘을 세번 수행



암호화: 암호-복호-암호

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



복호화: 복호-암호-복호

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

3중 DES

- 유효 키 길이가 168비트이므로, 슈퍼컴퓨터로도 5.3×10^{29}년이 걸려야 전수공격이 성공함
- 하지만 DEA보다 라운드 수가 3배나 많기 때문에 소프트웨어 구현속도가 느리며,

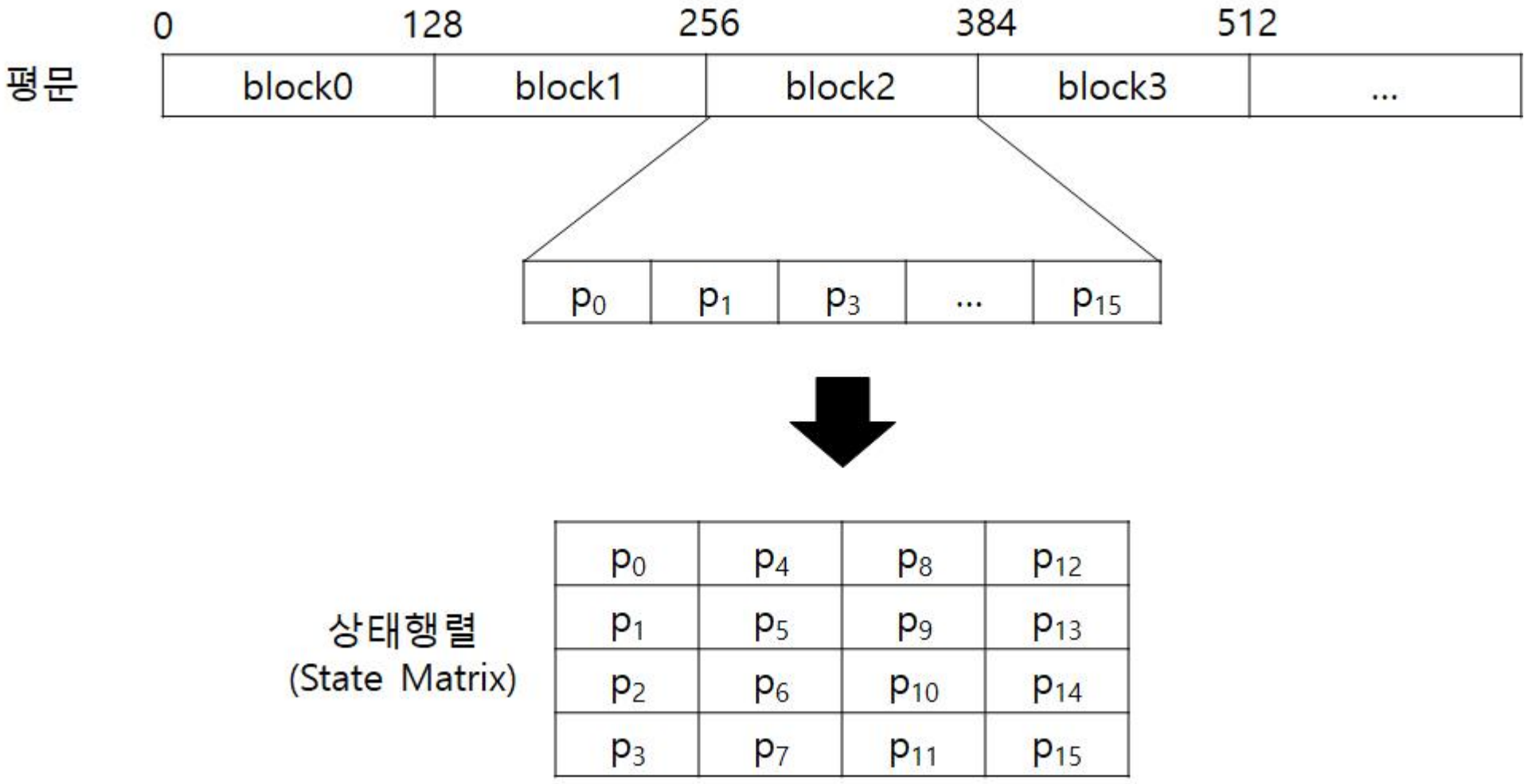
  64비트 블록을 사용하므로 보안이나 효율성이 떨어짐

-> 장기적으로 사용될 표준이 되기엔 부적합

AES(Advanced Encryption Standard)

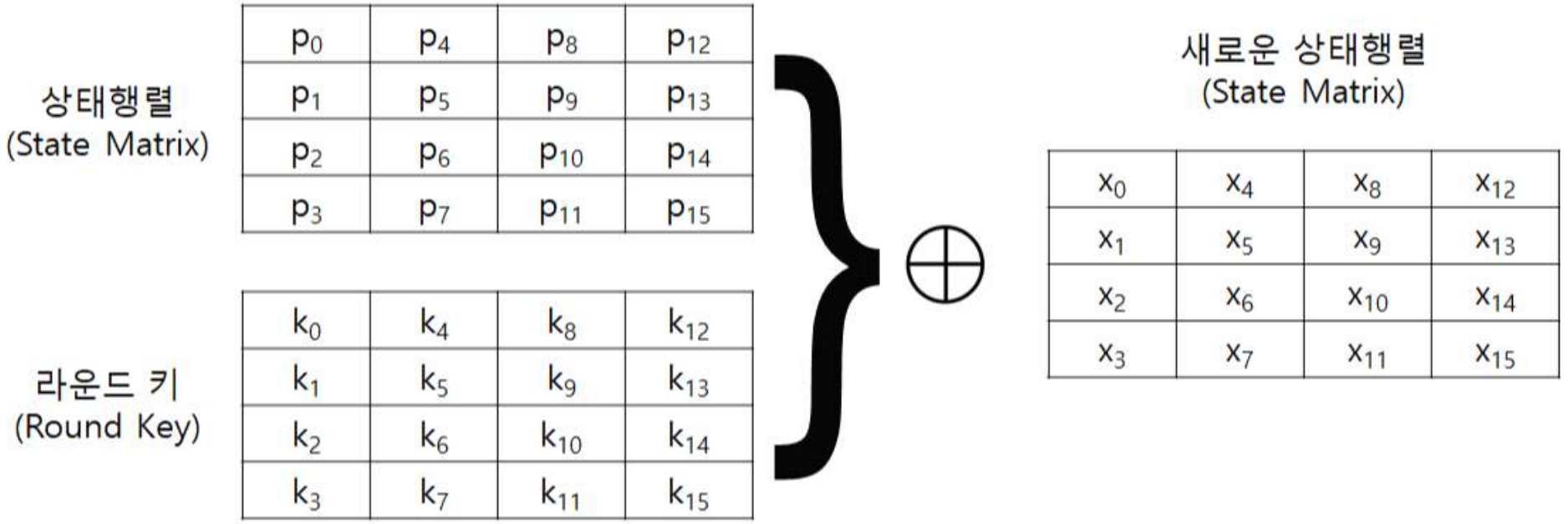
- 블록 길이: 128비트
- 키의 길이: 128, 192, 256비트
- 라운드 수: 10라운드
- 페이스텔 구조가 아님

AES 암호화


0
128
256
384
512
평문
block0
block1
block2
block3
...
p_0 p_1 p_3 ... p_{15}
상태행렬
(State Matrix)
p_0 p_4 p_8 p_{12}
p_1 p_5 p_9 p_{13}
p_2 p_6 p_{10} p_{14}
p_3 p_7 p_{11} p_{15}

AES 암호화

- 라운드 키는 Rotwords, SubByte, XOR 연산을 통해 키를 키 스케줄 워드로 확장



AES 암호화 라운드의 1단계

- 바이트 대체(Substitute bytes): S-box라는 표를 이용해 블록 교환

	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

0x19	0xA0	0x9A	0xE9
0x3D	0xF4	0xC6	0xF8
0xE3	0xE2	0x8D	0x48
0xBE	0x2B	0x2A	0x08

SubBytes

0xD4	0xE0	0xB8	0x1E
0x27	0xBF	0xB4	0x41
0x11	0x98	0x5D	0x52
0xAE	0xF1	0xE5	0x30

AES 암호화 라운드의 2단계

- 행 이동(Shift rows): 첫번째 위치부터 각 행의 위치가 증가되는 수만큼 각 행을 왼쪽으로 이동

0xD4	0xE0	0xB8	0x1E
0x27	0xBF	0xB4	0x41
0x11	0x98	0x5D	0x52
0xAE	0xF1	0xE5	0x30

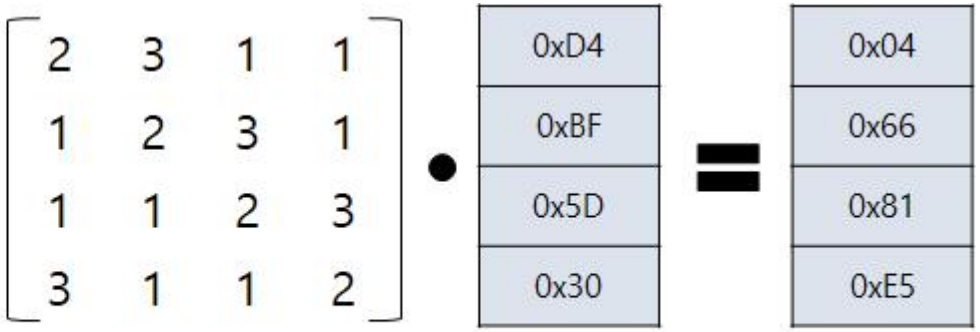
ShiftRows

0xD4	0xE0	0xB8	0x1E
0xBF	0xB4	0x41	0x27
0x5D	0x52	0x11	0x98
0x30	0xAE	0xF1	0xE5

AES 암호화 라운드의 3단계

- 열 섞기(Mix columns): 열을 고정 행렬과 섞음

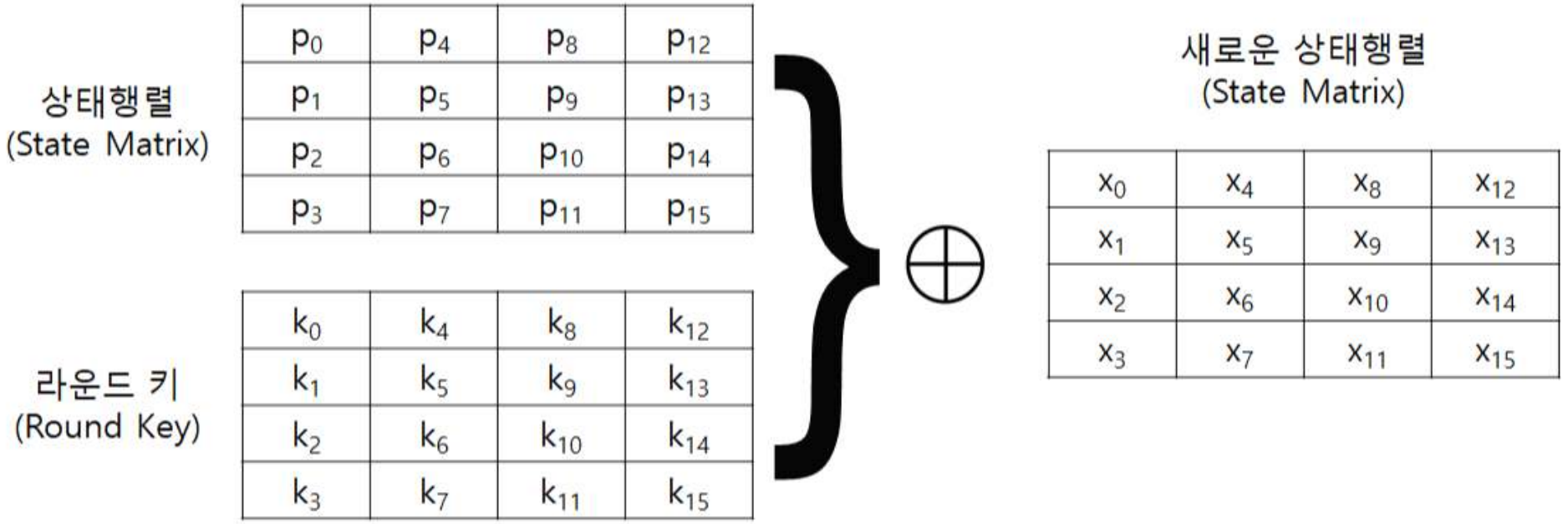
0xD4	0xE0	0xB8	0x1E
0xBF	0xB4	0x41	0x27
0x5D	0x52	0x11	0x98
0x30	0xAE	0xF1	0xE5



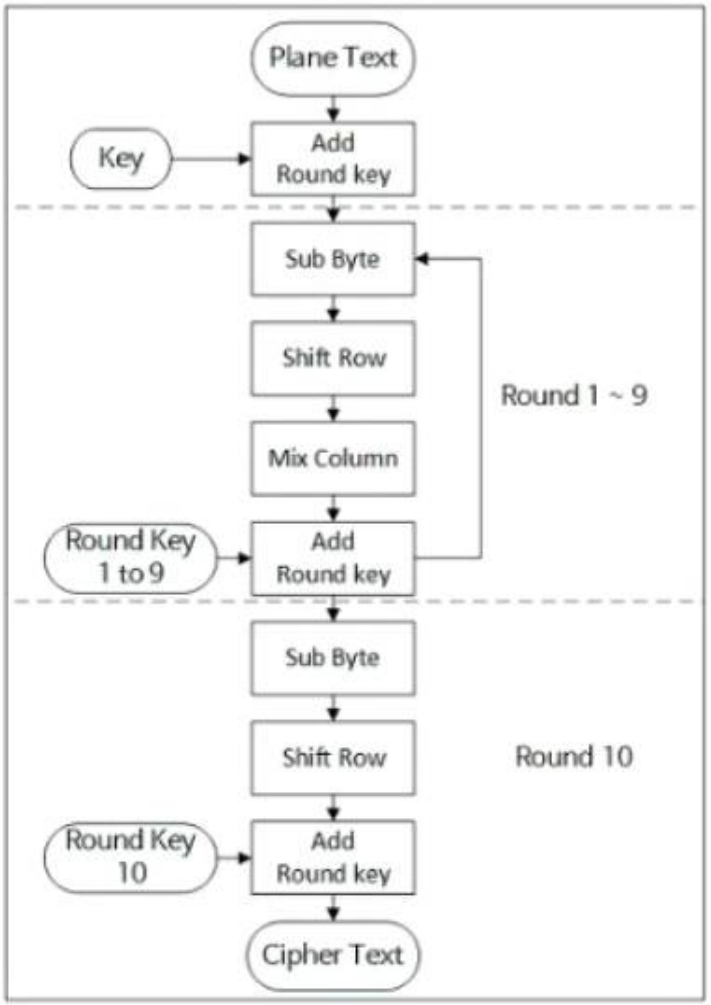
단, 마지막 10번째 라운드에서는 이 단계를 건너뜀

AES 암호화 라운드의 4단계

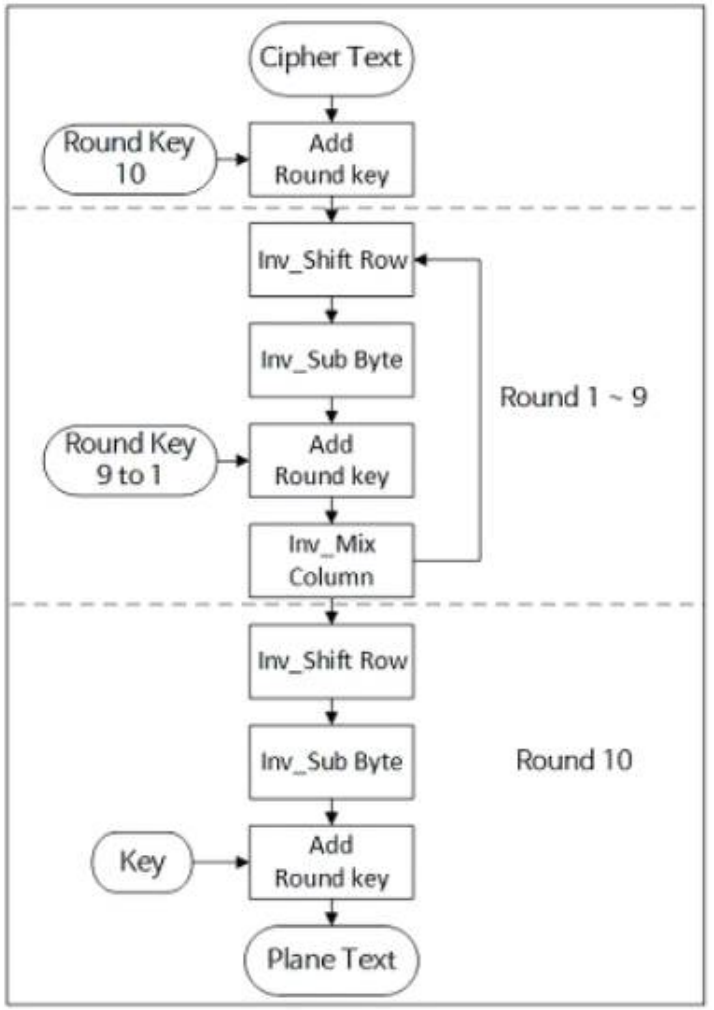
- 라운드 키 더하기(Add round key): 확장된 키의 일부와 현재 블록을 XOR한다



AES 암호화와 복호화



AES 암호화 알고리즘



AES 복호화 알고리즘

복호화:
A xor B xor B = A
라는 것을 이용하여 같은 라운드 키를 블록에 수행하여 역을 계산

AES의 장점

- 라운드 키 더하기와 다른 세 단계가 같이 작동하며 비트를 뒤섞어 보안성이 강화됨
- 역으로 계산하기 쉬움
- 키 길이가 128비트로 길어 전수공격의 위험이 없음

난수의 용도

- RSA공개 키 암호 알고리즘 및 기타 공개 키 알고리즘 키 생성에 이용됨
- 대칭 스트림 암호 스트림 키 생성에 사용
- 임시 세션 키 용도 대칭 키 생성시 사용
- Kerberos에서처럼 키를 배포해야 할 경우에 난수를 사용하여 재전송 공격을 막기 위한 핸드쉐이킹에 사용

난수 열의 두 가지 조건

- 무작위성
  - 균등 분포, 독립성을 충족해야함
  - 균등 분포의 경우에는 테스트하기 쉽지만, 독립성의 경우 여러 독립성 테스트를 통해 증명해야함
- 예측 불가성
  - 수열의 일부를 보고 이어지는 수를 예측할 수 없어야 함

의사난수

- 알고리즘 기법에 기초하여 생성된 난수
- 입력값에 의해 출력값이 결정되는, 진성 난수와 유사한 난수
- 통계적으로는 무작위성을 갖지 못함

-> 하지만 잘 만들어진 알고리즘이라면, 무작위성 테스트를 무사히 통과해 실제 사용에 문제가 없음

난수 생성기와 함수

- 진성난수 생성기(TRNG)

  - 엔트로피 소스(키 입력 타이밍 패턴, 디스크 전기 작용, 마우스 움직임, 시스템 클록의 순간 값 등)를 입력값으로 사용

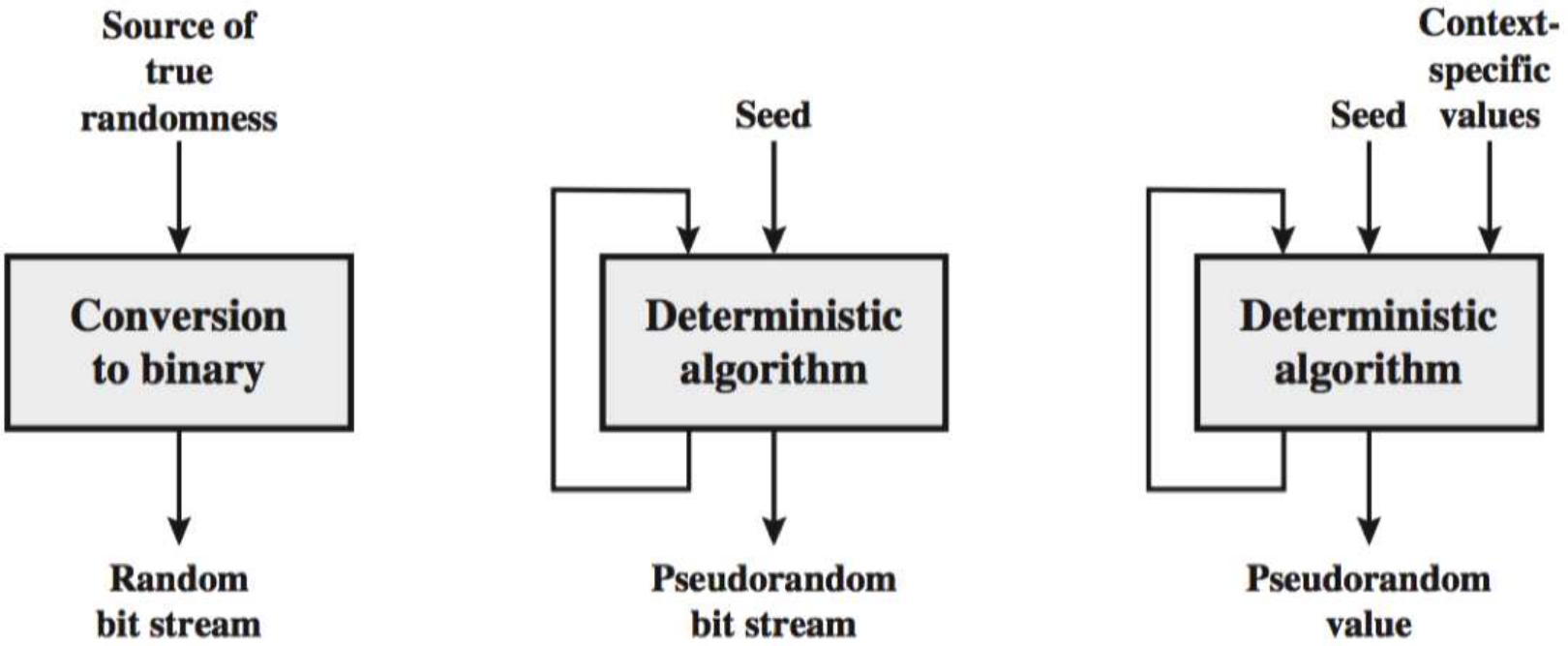
- 의사난수 생성기(PRNG)

  - 무한 비트열을 생성하기 위해 사용되는 알고리즘

- 의사난수 함수(PRF)

  - 고정된 길이 의사난수 비트열을 생성하기 위해 사용되는 함수

난수 생성기와 함수



TRNG = true random number generator
PRNG = pseudorandom number generator
PRF = pseudorandom function

Figure 7.1 Random and Pseudorandom Number Generators

의사난수 생성기의 알고리즘 범주

- **특정 목적 알고리즘**

- **기존 암호 알고리즘을 이용한 알고리즘**
  - 난수화된 입력 효과를 가지고 있음

Thanks!

강 민 채 (minchae@pel.sejong.ac.kr)