5 장

- ❖AES (Advanced Encryption Standard)
- ❖난수
- ❖기타 암호 알고리즘

새로운 암호

❑ DES의 brute-force공격에 대한 취약성 보완 필요

❖ 새로운 알고리즘 개발

➢ 상당한 암호학적 강도 유지

➢ 인터넷 기반 널리 응용

➢ DES 출현이후 개발된 현대 대칭블록 암호기술의 사용

❖ 차세대 암호 알고리즘 표준의 공모

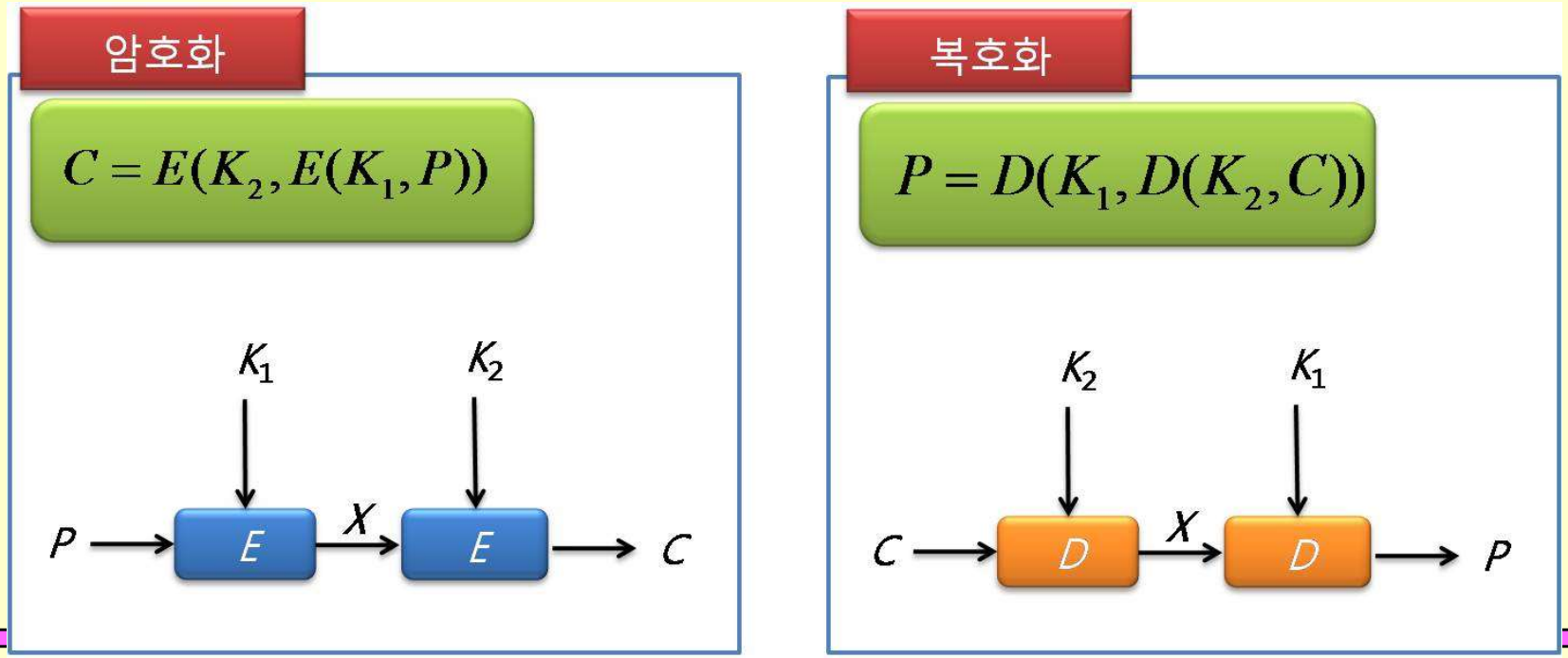
➢ AES

➢ 기존의 소프트웨어와 장비투자를 보전하기 위해 DES에 다중 키 사용 ; 2중 DES , 3중 DES

2중 DES

❑ 2개의 암호화 단계와 2개의 키 사용

- ❖ 가장 간단한 다중 암호방식
- ❖ 2개의 암호화 단계와 2개의 키를 가짐
- ❖ 56비트 키 2개를 이용하여 암호화 강도 증가
- ❖ 암호화 $C = E_{K_2}[E_{K_1}[P]]$, 복호화 $P = D_{K_1}[D_{K_2}[C]]$



< 2중 DES >

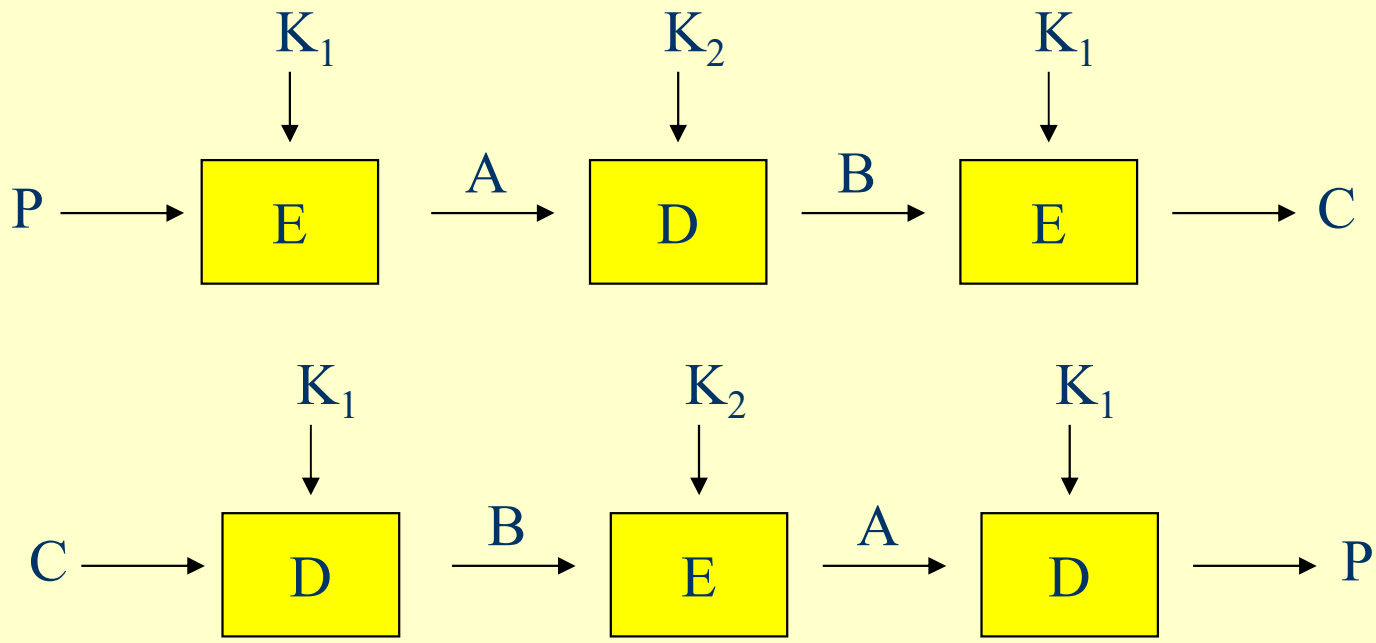
2중 DES

❑ 암호화 강도

- ❖ 56 비트 x 2개의 키 = 112 비트 길이의 키
- ❖ 단일키로의 축소
  - ➢ 키 K_3 존재에 대한 단일 단계로 축소: $E_{K_2}[E_{K_1}[P]] = E_{K_3}[P]$
- ❖ 중간결과에 의한 공격
  - ➢ Meet-in-the-middle Attack
  - ➢ DES의 특성 때문이 아니고, 블록 암호화에 대한 공격
  - ➢ 중간결과 : if $C = E_{K_2}[E_{K_1}[P]]$, $X = E_{K_1}[P] = D_{K_2}[C]$

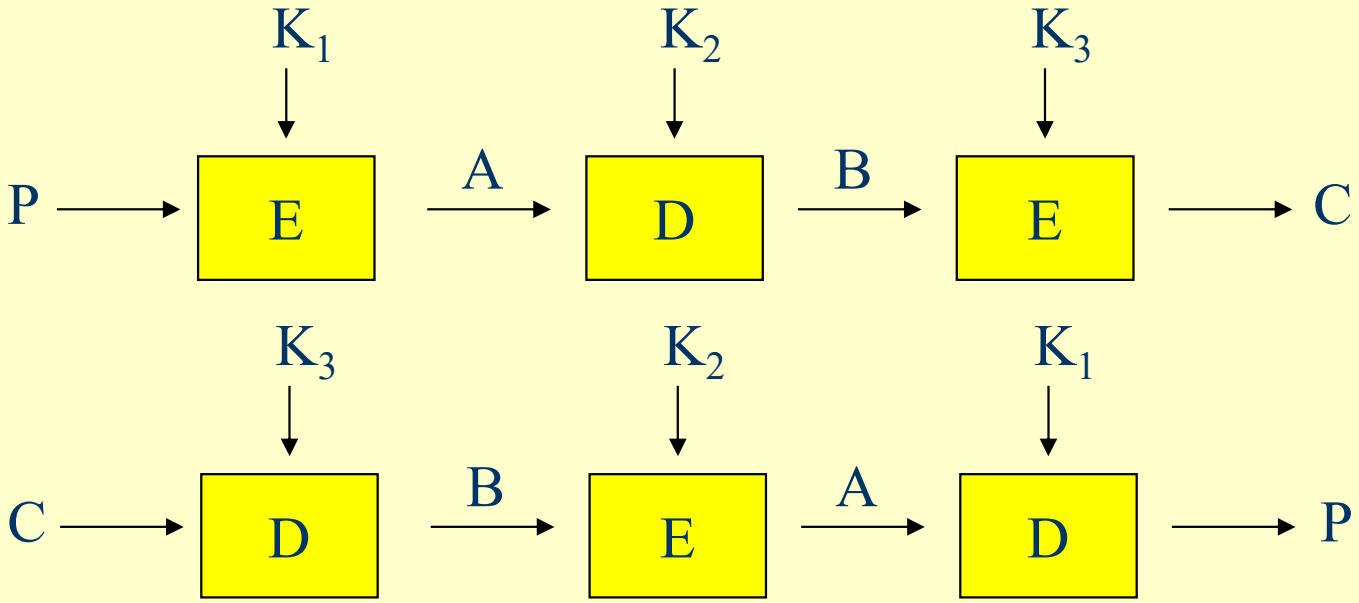
3중 DES

- 중간결과에 대한 공격 대책: 3단계 암호화
- 2키 3중 암호 방법 제안
  - 암호화 $C = E_{K_1}[D_{K_2}[E_{K_1}[P]]]$
  - 복호화 $P = D_{K_1}[E_{K_2}[D_{K_1}[C]]]$



3키에 의한 3중 DES

- 2키에 의한 3중 DES의 공격에 대한 우려사항
- 대안으로 [KALI96]: 3키 3중 DES을 제안
  - 56 x 3 =168비트 길이의 긴 키를 사용하는 단점
  - $C = E_{K_3}[D_{K_2}[E_{K_1}[P]]]$ $P = D_{K_1}[E_{K_2}[D_{K_3}[C]]]$
  - PGP, S/MIME, 인터넷 응용에서 사용



AES (Advanced Encryption Standard)

- DES의 취약성으로 인하여 3중 DES의 출현
  - DES(56비트 키)의 전사적 공격 취약성을 3중 DES의 128비트의 키 사용으로 보완
    - DES와 같은 알고리즘으로 동작
    - 전사적 공격 이외에는 어떠한 암호해독 공격 발견되지 않음
    - 보안에 대한 사항만 고려한다면 2010년까지는 암호 알고리즘의 표준으로 적합 (암호화 수행 속도 고려 않을 경우)
- 3중 DES의 단점
  - DES는 1970년대 중반 하드웨어 기반 구현을 목적으로 설계
    - 소프트웨어 상의 수행 속도가 느림
    - DES와 비교시 3배의 반복과정을 수행하므로 느림
  - 효율성과 보안성 측면에서 더 큰 블록 길이 요구
    - DES와 3중DES는 64비트 블록 길이 사용

❖ **Brute force Attack**

- ➢ 1970년대/80년대 : low computing power
- ➢ 90년대 : high computing power

Key Size (bits)	Number of Alternative Keys	Time Required at 1 Decryption/μs	Time Required at 10^6 Decryptions/μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31}\mu s = 35.8$ minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\mu s = 1142$ years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127}\mu s = 5.4 \times 10^{24}$ years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167}\mu s = 5.9 \times 10^{36}$ years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26}\mu s = 6.4 \times 10^{12}$ years	6.4×10^6 years

AES(Advanced Encryption Standard)의 역사

- 3중DES를 대체하기 위해 개발
- 1997년 NIST는 AES 암호 알고리즘 공모
- 평가기준의 3가지 범주
  - ❖ 보안성
    - ➢ 알고리즘 해독에 요구되는 노력
    - ➢ AES의 최소 길이가 128bit 이므로 전사적 공격은 고려될 필요 없음
      128bits : $5.4 * 10^{18}$ years (10^6 Encryption per micro sec)
  - ❖ 비용
    - ➢ NIST는 AES가 넓은 응용분야에 적용될 것을 의도
    - ➢ 높은 계산 효율성과 광대역 링크와 같은 고속 응용에 적합해야 함
  - ❖ 알고리즘 및 개발특성
    - ➢ 유연성, 다양한 하드웨어와 소프트웨어에 대한 적합성
    - ➢ 알고리즘의 단순성 등의 고려사항 포함
  - ❖ AES는 효율, 보안, 성능, 구현, 유연성 면을 고려

AES의 역사

- 이상의 기준을 적용하여 후보알고리즘이 21개→15개→5개로 압축
  - 1차 평가: 15개 알고리즘 채택하여
  - 1999년 4월경 AES 후보 알고리즘은 모두 5개 (MARS, RC6, Rijndael, Serpent, Twofish)
  - 컨퍼런스를 거쳐 2000년 10월 Rijndael이 선정
  - 2001년 11월 : Rijindael을 최종 표준으로 채택 ; [FIBSPUB 197] 공포
    - 벨기에 출신 암호학자 John Daemen 박사, Vincent Rijmen 박사에 의해 개발
    - Rijndael (발음, Rain-dal 혹은 Regin-dal)

AES의 역사

- 미국의 차세대 암호 표준으로 2001년 NIST에서 AES를 공포
  - 5년의 표준화 과정을 거쳐 2001년 11월 26일에 연방 정보 처리 표준(FIPS 197)으로 발표
  - 2002년 5월 26일부터 표준으로 효력을 발휘하기 시작
  - DES를 대신할 대칭블록 암호방식으로 인정
  - 3중DES이상의 보안 안전성과 향상된 효율성의 요구사항 만족
  - 128비트 대칭블록암호화 방식이고, 128, 192, 256 비트 길이의 키를 지원 명시

- 적어도 향후 20~30년 정도까지는 안전성이 보장된 차세대 암호화 표준
  - 당분간 3중 DES 공인 알고리즘 지속(미국 정부용)
  - 3중 DES가 AES로 대체되는데 몇 년 소요 예측

AES 평가

❑ 최종 개량된 평가기준

❖ 일반적인 보안성

- 3년의 평가기간 동안 암호학계에서 수행된 공개 보안성 분석
- 비 선형 구조로 S-box 사용

❖ 소프트웨어구현

- 실행속도, 다양한 플렛폼에서의 성능, 키 길이에 대한 속도 변화
- 8비트 및 64비트의 다양한 플렛폼에서 암복화를 수행
- 뛰어난 병렬 구조는 프로세스자원을 효율적으로 이용

❖ 제한된 공간 환경

- 스마트 카드와 같은 ROM,RAM이 제한된 응용에서 사용 가능성
- 암복화 하나의 구현시 제한 된 환경에 적합(동시 구현시 ROM의 요구도 증가)

AES 평가

❑ 최종 개량된 평가기준

❖ 하드웨어 구현

- ➢ 소프트웨어 구현과 비교하여 속도, 크기, 구현비용에 최적화 여부
- ➢ 하드웨어에서 메모리가 커지면 가격도 배 이상이 되는 문제 있음
- ➢ 완전한 파이프라인 구현의 경우 공간 요구도는 증가하지만, 성능에는 영향 없음

❖ 암호 대 복호

- ➢ 암복화 알고리즘이 서로 다르면 복호 알고리즘을 위한 추가 공간이 필요
- ➢ 암호화와 복호화 함수가 서로 다름
- ➢ 동시 구현은 암호화 구현 보다 60% 공간이 더 필요(속도는 큰 차이가 없음)

AES 평가

- 최종 개량된 평가기준
  - ❖ 키의 민첩성
    - ➢ 키를 얼마나 빠르게, 최소의 자원으로 서브키로 변환 및 교환 능력
    - ➢ 서브 키 계산이 매우 빠름(키 스케쥴을 단 한번에 실행)
  - ❖ 기타 융통성 및 유연성
    - ➢ 키와 블록 사이즈에 대한 지원의 용이성
    - ➢ 새로운 공격에 대응하는 라운드 횟수의 증가 용이성
    - ➢ 특수한 환경에서 암호 요소들을 최적화 할 수 있는 유연성
    - ➢ 128,192 및 256 비트의 키 길이와 블록 크기를 제공
    - ➢ 라운드 회수 변화 뿐만 아니라 32 배수의 어떤 키와 블록 크기도 지원

Rijndael AES 암호

- Rijndael 알고리즘
  - 키 길이: 128bit, 192bit, 256bit 중 선택
  - 블록길이: 128bit로 제한
- Rijndael 특성
  - 모든 알려진 공격에 대한 저항력
  - 다양한 플랫폼에 대한 속도와 코드의 간결성
  - 단순한 설계
- AES 파라미터

키 길이	128 bit	192 bit	256 bit
평문 블록 사이즈	128 bit	128 bit	128 bit
라운드 수	10 회	12 회	14 회
라운드 키 길이	128 bit	128 bit	128 bit
확장 키 길이(워드/바이트)	44/176	52/208	60/240

SEED

- 1999년 2월 한국정보보호진흥원이 개발한 128비트및 256비트 대칭 키 블록 암호 알고리즘
- 민간 부분인 인터넷, 전자상거래, 무선 통신 등에서 공개 시에 민감한 영향을 미칠 수 있는 정보의 보호와 개인 프라이버시 등을 보호하기 위하여 개발된 블록암호알고리즘
  - 블록 크기 : 128 비트 (16 바이트)
  - 키 크기 : 128 비트 (16 바이트)
  - 구 조 : Feistel Network
  - 라운드 수 : 16

- 우리나라 암호 알고리즘
  - ARIA, HIGHT(경량화 알고리즘)
- 서명 알고리즘
  - KCDSA, EC-KCDSA

난수의 사용

- 난수(random number)
  - 특정한 배열 순서나 규칙을 가지지 않는 연속적인 임의의 수
- 의사 난수(pseudo random number)
  - 컴퓨터에 의해서 만들어지는 난수
  - 아주 긴 주기를 가지고 있는 숫자 열
- 난수의 사용 예
  - 비표(nonce) : 재전송 공격 방지나 블록 암호의 CTR 모드
  - 세션 키
  - 공개키 암호 알고리즘을 위한 키 생성
  - 대칭 스트림 암호를 위한 비트 스트림 생성
  - 초기화 벡터(IV) 생성 : 블록 암호 모드인 CBC, CFB, OFB

난수의 사용

- 난수의 요구사항
  - 임의성(randomness) : 통계학적 관념상 수의 순서가 임의적이어야함
    - 「아무렇게」보이는 성질
    - 의사난수열의 통계적인 성질을 조사해서 치우침이 없도록 하는 성질
      - 균일 분포 : 수열의 비트 분포가 균일해야 함
      - 독립성 : 수열의 어느 부분 수열도 다른 부분 수열로 부터 추정이 될 수 없어야 함
  - 비예측성(unpredictability) : 수열의 잇따른 다음 수의 순서에 대해 예측이 불가능해야 함
    - 과거에 출력한 의사 난수열이 공격자에게 알려져도 다음에 출력하는 의사난수를 공격자는 알아맞힐 수 없다는 성질
    - 알고리즘은 공격자에게 알려져 있다고 가정하고 Seed를 사용
      - 시드 (seed) : 공격자에게 비밀

난수의 사용

❑ 난수의 요구사항

❖ 재현 불가능성(reconstruction is impossible) : 같은 수열을 재현할 수 없다는 성질

- ➢ 재현하기 위해서는 그 난수열 자체를 보존해두는 것 이외에 방법이 없는 성질
- ➢ 소프트웨어만으로는 재현 불가능성을 갖는 난수열 생성 불가
- ➢ 소프트웨어는 의사난수열만 생성가능

❖ 주기(period) : 반복이 다시 시작할 때 까지의 수열의 길이

- ➢ 소프트웨어가 생성하는 수열은 언젠가는 반복
- ➢ 주기를 갖는 수열은 재현 불가능하지 않음

난수의 사용

❑ 재현 불가능한 난수 생성

❖ 재현 불가능한 물리 현상으로부터 정보를 취득

- ➢ 주위의 온도나 소리의 변화
- ➢ 사용자의 마우스 위치 정보
- ➢ 키스트록 입력 시간 간격
- ➢ 방사선 관측기의 출력 (다양한 하드웨어로부터 얻어진 정보)

❑ 진성난수(Real Random Number)

❖ 재현 불가능한 난수

❖ 무작위성, 예측 불가능성, 재현 불가능성을 가짐

예) 동전 던지기

난수의 사용

	무 작위성	예측 불가능성	재현 불가능성	비고	암호기술 적용 여부
약한 의사 난수	O	X	X	무작위성만 갖는다	암호 기술에 사용 불가
강한 의사 난수	O	O	X	예측 불가능성도 갖는다	암호 기술에 사용
진정한 난수	O	O	O	재현 불가능성도 갖는다	암호 기술에 사용

난수의 사용

❑ 난수의 용도

- ❖ 아무리 강한 암호 알고리즘이라도 키가 공격자에게 알려져 버리면 아무 의미가 없다
- ❖ 난수를 사용해서 키를 만들어, 공격자에게 키를 간파당하지 않도록 하는 것
- ❖ 난수를 사용하는 목적은 **간파당하지 않도록 하기 위한 것**

특성	의사 난수 생성기	진정한 난수 생성기
효율성	매우 휼륭함	나쁨
결정론	결정론적 특성	비결정론적 특성
주기성	주기성을 띔	비주기성을 띔

의사 난수 생성의 원리

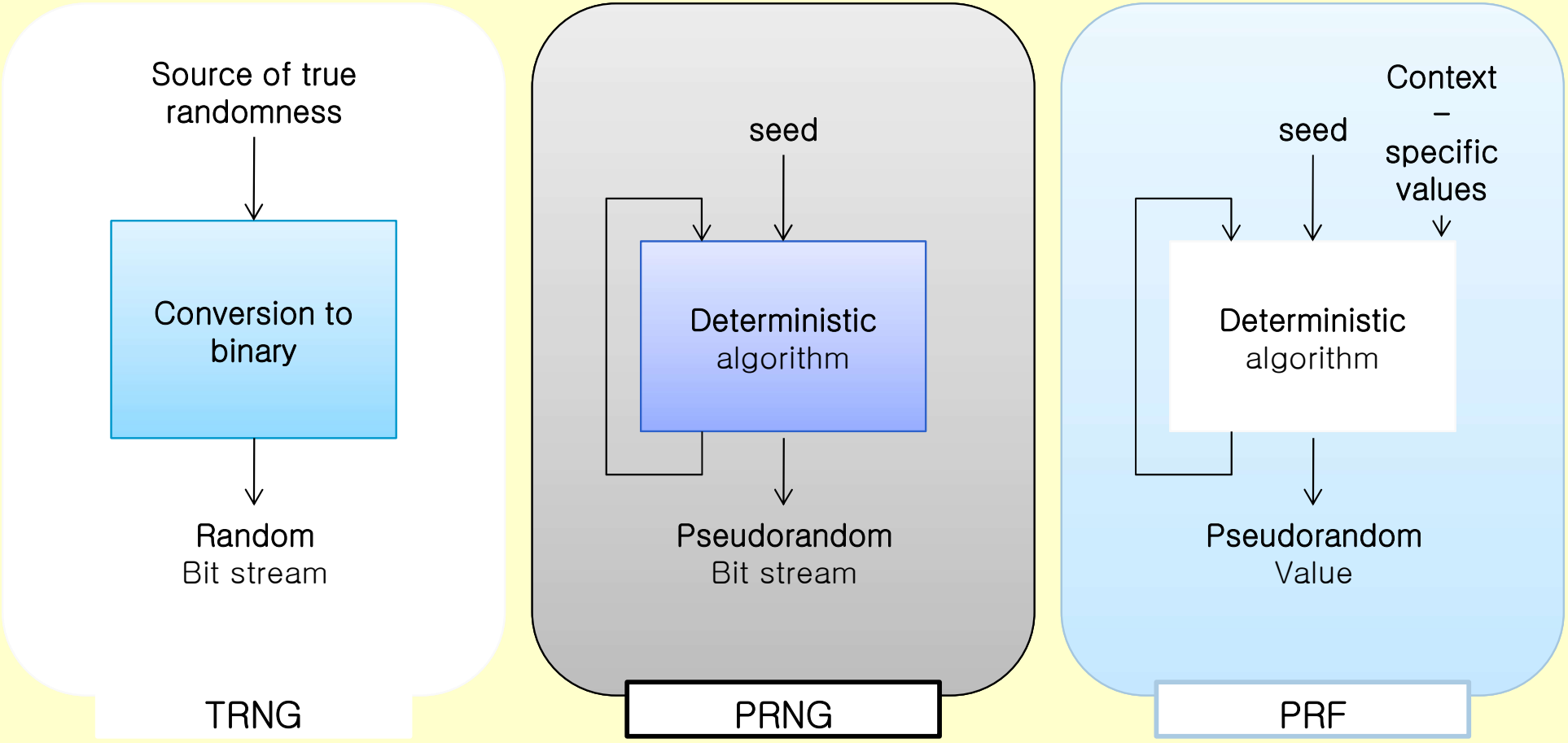
- TRNG(True Random Number Generator : 진 난수 생성기)
  - 실제적으로 랜덤한 소스를 입력으로 사용
    - 키보드 입력 타이밍 패턴 및 마우스 움직임
    - 디스크의 전기적 활동, 시스템 클럭의 순간 값

- PRNG(Pseudo Random Number Generator : 의사난수발생기)
  - 고정값 seed를 입력받아 결정적 알고리즘을 사용하여 출력 비트열 생성
  - 제한이 없는 비트열 생성하는데 사용
    - 알고리즘과 시드를 알고 있는 공격자는 비트열 재생성 가능

- PRF(Pseudo Random Function : 의사난수함수)
  - 고정된 길이의 의사 난수 비트열을 생산하는데 사용

의사 난수 생성의 원리

❑ 난수와 의사난수 생성기



의사 난수 생성의 원리

❑PRNG 요구 사항

- ❖ Seed를 알지 못하는 공격자가 의사 난수열을 결정할 수가 없어야 함

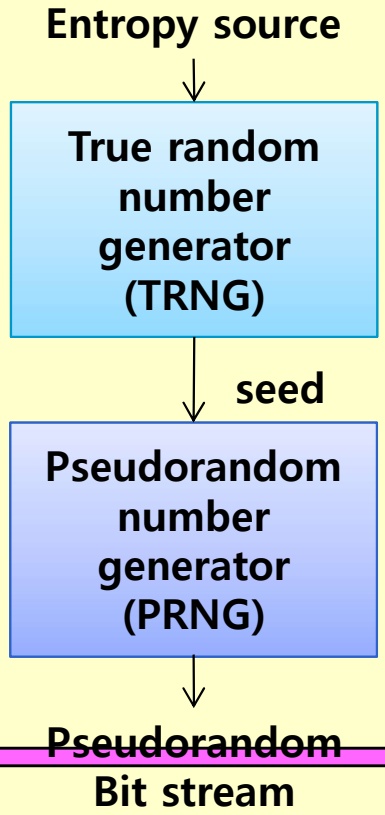
- ❖ 임의성(Randomness)
  - ➢ 생성된 비트 스트림이 결정적일지라도 랜덤하게 보여야 함
  - ➢ 균일성
    - 난수 또는 의사 난수 비트열의 생성에 있어서 0과 1은 거의 동일하게 존재
  - ➢ 확장성
    - 비트열이 랜덤하면 무작위로 추출된 어떤 비트열도 랜덤해야 함
  - ➢ 일관성
    - 생성기의 동작은 초기값 전반에 대해 일관되어야 함

의사 난수 생성의 원리

❑PRNG 요구 사항(계속)

- ❖ 비예측성(Unpredictability)
  - ➢ 수열의 잇따른 다음수의 순서에 대해 예측이 불가능해야 함
  - ➢ 전 방향 비예측성
    - 이전 비트들에 대한 정보가 있다고 하더라도 다음 출력 비트는 예측할 수 없어야 함
  - ➢ 후 방향 비예측성
    - 생성된 어떠한 값의 정보를 통해서도 seed를 결정할 수 없어야 함

- ❖ 시드요구사항(Seed Requirements)
  - ➢ Seed는 예측 불가능해야 함
  - ➢ Seed는 난수 또는 의사난수이어야 함
  - ➢ TRNG에 의해 seed생성



의사 난수 생성기

- 선형합동생성기
  - Lehmer에 의해 처음 제안된 알고리즘
  - 선형 합동 방법

m	법(modulus)	m>0
a	승수(multiplier)	0<a<m
c	증분(increment)	0≤c<m
X_0	초기치 혹은 seed	$0 \le X_0 < m$

$$X_{n+1} = (aX_n + c) \bmod m$$

  - a, c및 m값의 선정은 좋은 난수 생성기의 개발에 중요
  - 난수 생성기의 평가 기준
    - T1 : 난수 생성 함수는 최대 생성 주기를 가져야 한다. 즉, 함수는 반복되기 전에 0과 m사이의 모든 값을 생성해야 한다.
    - T2 : 생성된 수열은 랜덤하게 보여야 한다.
    - T3 : 함수는 32비트 연산을 효율적으로 수행해야 한다.

의사 난수 생성기

- **선형합동생성기**

m	법(modulus)	m>0
a	승수(multiplier)	0<a<m
c	증분(increment)	0≤c<m
X_0	초기치 혹은 seed	$0 \le X_0 < m$

$$X_{n+1} = (aX_n + c) \bmod m$$

- a= c = 1 X_0 = 5 6,7,8,9…
  - X1 = (a X0 +c) mod m = 1*5 + 1 = 6
  - X2 = (a X1 +c) mod m = 1*6 + 1 = 7
  - X3 = (a X2 +c) mod m = 1*7 + 1 = 8

- a=7, c=0, m=32, X_0 = 1 ➔ 7,17,23,1,7
  - X1 = (a X0 +c) mod m = 7*1 + 0 = 7
  - X2 = (a X1 +c) mod m = 7*7 + 0 = 49 mod 32 = 17
  - X3 = (a X2 +c) mod m = 7*17 + 0 = 119 mod 32 = 23
  - X4 = (a X3 +c) mod m = 7*23 + 0 = 161 mod 32 = 1

- a=5, c=0, m=32, X_0 = 1 ➔ 5,25,29,17,21,9,13,1

의사 난수 생성기

❑ Blum Blum Shub 생성기

- ❖ 개발자들의 이름에서 유래 [BLUM86]
- ❖ 어떠한 특정 목적의 알고리즘에서도 암호학적 강도를 증명하는 가장 강력하게 통용되는 수단
- ❖ 암호학적으로 안전한 의사난수 비트 생성기로 불림
  - ➢ CSPRBG : Cryptographically Secure Pseudo Random Bit Generator
- ❖ n의 소인수 분해 문제에 대한 어려움에 기반
  - ➢ n이 주어졌을 때, n의 두 소수 인수 p와 q를 알아야 함

$p \equiv q \equiv 3 (\mathrm{mod}\ 4)$

$n = p \times q$

$gcd(n, s)=1$, s선정(서로소)

$X_0 = S^2 \bmod n$

for i = 1 to ∞

$X_i = (X_{i-1})^2 \bmod n$

$B_i = X_i \bmod 2$

의사 난수 생성기

$X_0 = S^2 \bmod n$

for i = 1 to ∞

$X_i = (X_{i-1})^2 \bmod n$

$B_i = X_i \bmod 2$

❖ **BBS 생성기의 연산 예**

- p=7 q=11 n = 7*11 =77
- seed s = 13 GCD(77 13) = 1

- $X_0 = S^2 = 13^2 = 169$
  $= 15 \bmod 77$

- $X_1 = (X_0)^2 = 15^2 = 225$
  $= 71 \bmod 77$
- $B = 71 \bmod 2 = 1 \bmod 2$

- $X_2 = (X_1)^2 = 71^2 = 5041$
  $= 36 \bmod 77$
- $B = 36 \bmod 2 = 0 \bmod 2$

i	X_i	B_i
0	15	
1	71	1
2	36	0
3		
4		
5		
6		
7		
8		
9		
10		

의사 난수 생성기

❖ **BBS 생성기의 연산 예**

- $n = 192648 = 383 \times 503$
- $seed\ s = 101355$

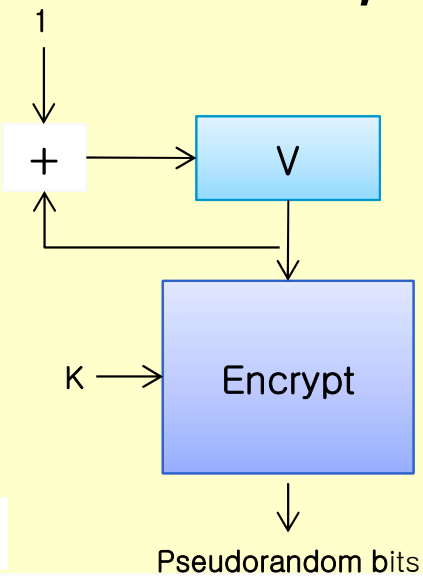
i	X_i	B_i
0	20749	
1	143135	1
2	177671	1
3	97048	0
4	89992	0
5	174051	1
6	80649	1
7	45663	1
8	69442	0
9	186894	0
10	177046	0

I	X_i	B_i
11	137922	0
12	123175	1
13	8630	0
14	114386	0
15	14863	1
16	133015	1
17	106065	1
18	45870	0
19	137171	1
20	48060	0

블록 암호를 사용한 의사 난수 생성

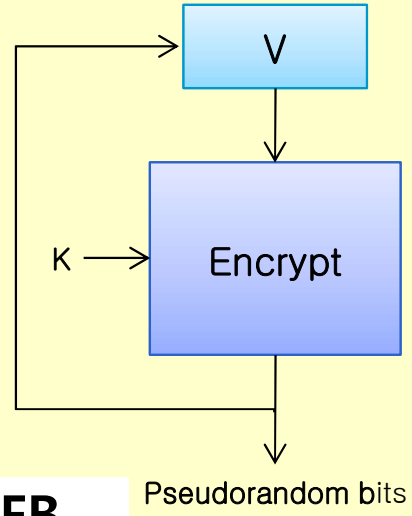
❑ 블록 암호 운용 모드를 이용한 PRNG

- ❖ CTR모드 : SP800-90, ANSI 표준 X9.82(난수 생성), RFC4086
- ❖ OFB모드 : X9.82, RFC4086



CTR

while (len(temp)
< requested_number_of_bits) do
V = (V + 1) mod 2^{128}
output_block = E(Key, V)
temp = temp || output_block

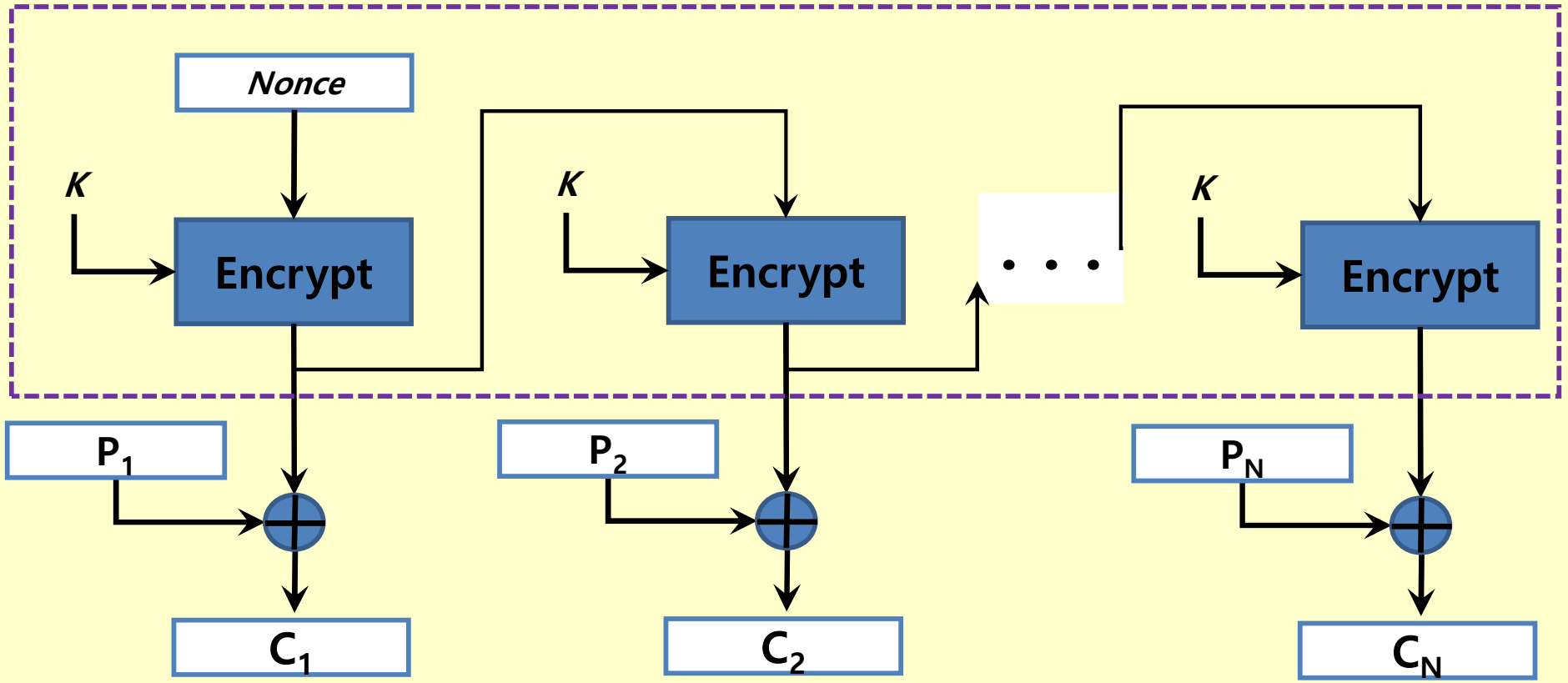


OFB

while (len(temp)
< requested_number_of_bits) do
V = E(Key, V)
temp = temp || V

출력 피드백 모드

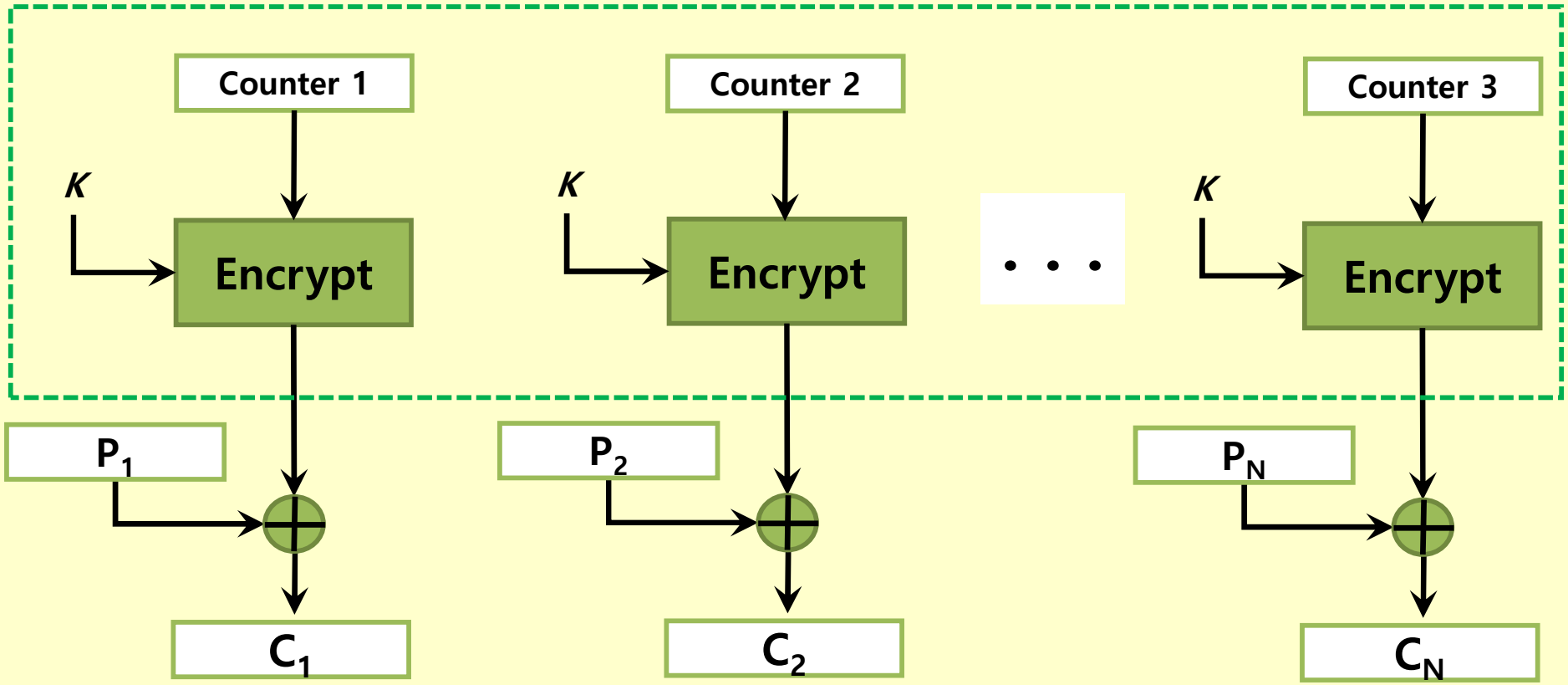
❑ 출력피드백(OFB) 모드



Encryption

계수기 모드

❑ 계수기(CTR) 모드

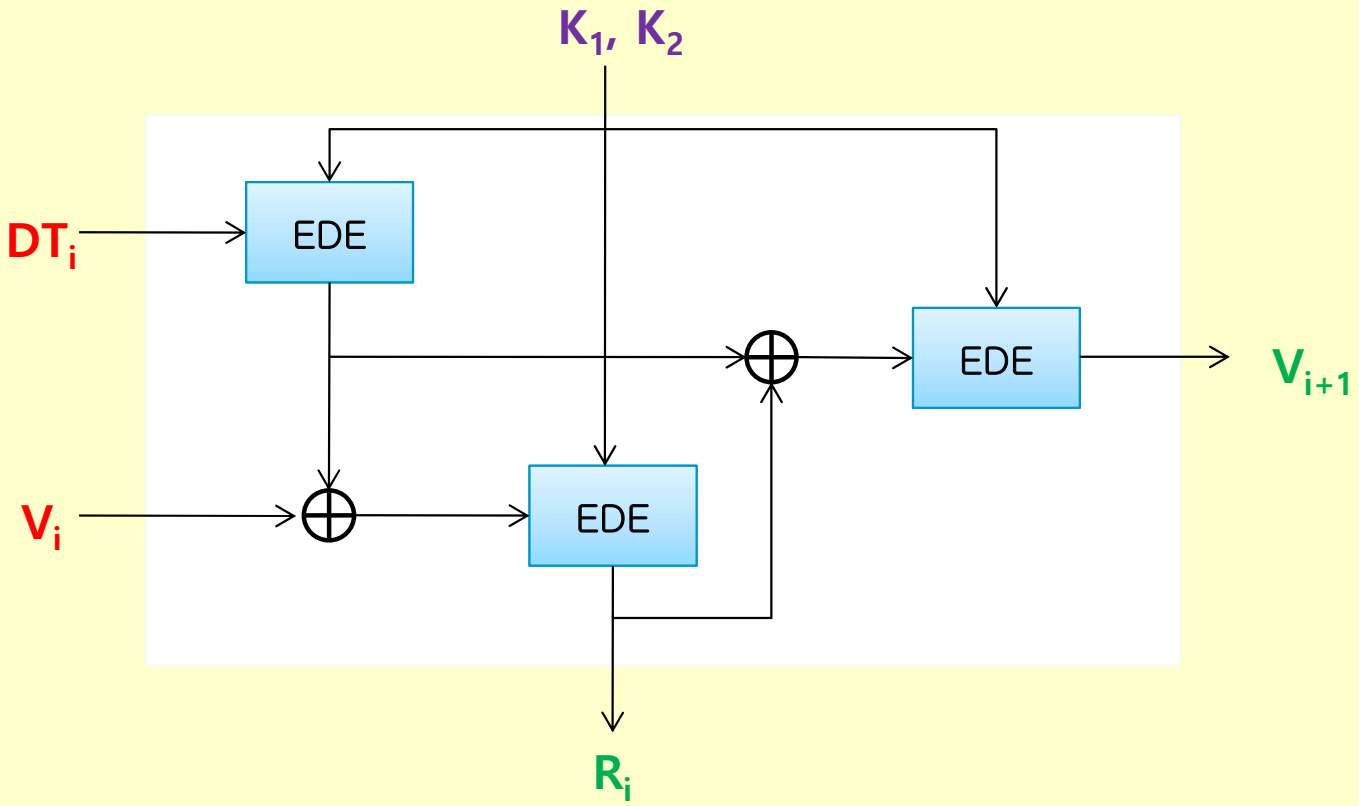


Encryption

블록 암호를 사용한 의사 난수 생성

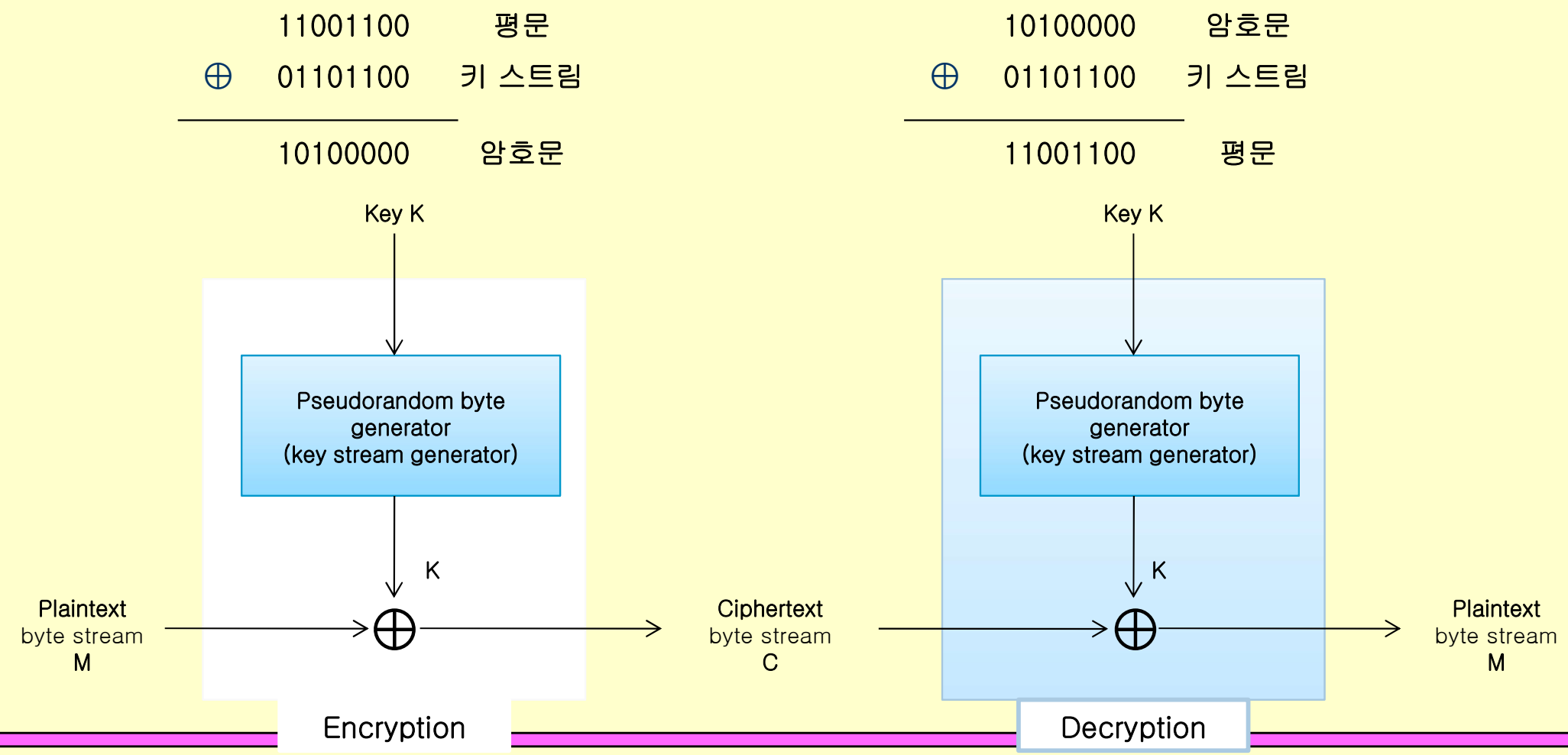
❑ ANSi X9.17 의사 난수 생성기

- ❖ 입력 : 현재의 날짜와 시간(64비트), seed값
- ❖ 키 : 3개의 3-DES모듈 사용, 동일한 56비트 키 쌍 사용
- ❖ 출력 : 64비트 의사 난수와 64비트 seed값으로 구성



스트림 암호

❑ 일반적인 스트림 암호는 한번에 1 바이트씩 평문 암호화



스트림 암호

- 스트림 암호 설계 시 고려사항
  - 반복 주기가 길어야 함
  - 키 스트림이 랜덤하게 나타나야 함
  - 입력 키가 충분히 길어야 함
- 스트림 암호의 장점 및 단점
  - 동일한 크기의 키를 사용하는 블록 암호만큼 안전함
  - 블록 암호에 비해 간단하고 빠른 처리 속도
  - 동일한 키를 가지고 두 개 이상의 평문을 암호화 할 경우, 해독이 간단해질 수도 있음(재사용이 안됨)

Cipher	Key Length	Speed(Mbps)
DES	56	9
3DES	168	3
RC2	Variable	0.9
RC4	Variable	45

국제 데이터 암호 알고리즘
(IDEA : International Data Encryption Algorithm)

- 스위스 연방 기술 연구소의 Xueja Lai와 James Massey에 의해 1990년에 개발
- 초기 PES(Proposed Encryption Standard)는 이후 1992년 IDEA(International Data Encryption Algorithm)로 이름을 고쳐 제안
- DES를 대체하기 위해 제안된 관용 암호 알고리즘 중 하나
- **유럽 표준**
- E-mail 암호를 위한 **PGP**에 포함
- 설계원리
  - 64비트 블록의 데이터 입력
  - 암호화와 복호화에 동일한 알고리즘
  - 128비트의 키를 사용
  - 블록 암호
  - 반복횟수 : 전체 8라운드
  - 하드웨어나 소프트웨어로 쉽게 구현

❑ 암호학적 강도

- ❖ 블록길이
  - ➢ 통계적 분석을 막을 수 있을 만큼 길어야 함
  - ➢ 64비트의 블록이면 충분
- ❖ 키의 길이
  - ➢ 모든 키의 탐색(Exhaustive Search)을 효율적으로 막을 수 있을 만큼 커야 함
  - ➢ 128비트면 향후에도 안전할 것이라고 여겨짐
- ❖ 혼돈
  - ➢ 목적 : 암호문의 통계적 성질이 평문의 통계적 성질에 의존하는지에 대한 결정을 복잡하게 만드는 것
  - ➢ 세가지 연산 : XOR, 덧셈, 곱셈
- ❖ 확산
  - ➢ 목적 : 각 평문 비트는 모든 암호문 비트에 영향을 끼쳐야 하고, 각 키 비트는 모든 암호문 비트에 영향을 주어야 함

❑ 구현상의 고려 사항

❖ 소프트웨어 구현을 위한 설계 원칙

- ➢ 서브블록의 사용
  - 암호연산은 소프트웨어에 대해 당연히 8, 16, 32비트와 같은 서브 블록에서 동작하도록 한다
  - IDEA는 16비트 서브 블록을 사용
- ➢ 간단한 연산의 사용
  - 덧셈, 자리 이동 등을 사용하여 쉽게 프로그램 되어야 함
  - IDEA의 기본 연산은 이 요구사항을 만족

❖ 하드웨어 구현에 대한 설계 원칙

- ➢ 암호화 복호화의 유사성
  - 암호화와 복호화는 키를 사용하는 방법에서만 달라야 함
- ➢ 정규구조
  - VLSI 구현을 용이하게 하기 위한 정규적인 모듈 구조를 가져야 함

Blowfish

- Bruce Schneier에 의해 개발된 대칭 블록 암호 방식
- 1993년에 만들어져 현재는 비특허로 모든 사용자에게 무료로 개방
- 특성
  - 빠른 속도 : 32비트 마이크로 프로세스에서 1 바이트 당 18클럭 사이클의 속도로 암호화
  - 간결성 : 5K 이내의 메모리에서도 실행가능
  - 단순성 : 간단한 구조로 구현이 쉽고 알고리즘의 강도 결정이 용이함
  - 보안의 가변성 : 키의 길이는 가변적이며 448비트만큼 길어질 수 있어 높은 속도와 보안성 사이의 균형이 가능
- 64비트 블록의 평문을 암호화
- 키의 길이가 32~448비트인 가변 길이 키를 사용하는 비밀 키 블록 암호
- Much faster than DES and IDEA
- Unpatented and royalty-free
- Free source code available

RC5

❑ RC5

- ❖ RC5는 1994년 RSA Security사의 Ronald Rivest에 의해 고안된 블록 방식의 알고리즘
- ❖ 다양한 크기의 키, 블록, 라운드를 가짐
- ❖ 라운드도 가변적
- ❖ 블록의 크기는 32, 64, 128 비트
- ❖ 키의 크기는 0에서 2040 비트까지 가변적으로 사용
- ❖ 라운드는 0에서 255까지 가변적
- ❖ 효율과 보안성을 고려하여 사용자가 원하는 만큼 성능과 안전성을 위해 수치를 조절

❑ RC6

- ❖ RC5를 기반으로 하여 Rivest, Sidney, Yin에 의해 재설계된 블록 암호화 방식의 알고리즘
- ❖ RC5와 거의 동일한 방식
- ❖ 목적 : AES의 요구 사항을 만족하여 미국내 표준으로 제정되는 것
- ❖ AES 결정을 위한 최종 5개 후보중의 하나
- ❖ RC6가 RC5와 다른점은 2가지
  - ➢ 정수 곱셈방식이 추가
  - ➢ RC5의 2비트 레지스터 대신에 4비트의 작동레지스터를 사용
- ❖ 현재 RC6는 특허 등록이 안 되어 있고, 무료로 사용할 수 있는 알고리즘

CAST-128

- 1997년 Carlisle Adams와 Stafford Tavares에 의해 개발된 대칭 암호 알고리즘
- 40비트에서 128비트 사이에 8 비트씩 증가하는 다양한 크기의 키를 사용
- 64비트의 평문블록 => 64비트의 암호문블록
- 16회 반복하는 고전적 Feistel 네트워크 구조
- 각 반복 과정에 두개의 서브키를 사용