제 13 장
난 수

박 종 혁 교수

Tel: 970-6702

Email: jhpark1@seoultech.ac.kr

1절 난수가 사용되는 암호 기술

2절 난수의 성질

3절 의사난수 생성기

4절 구체적 의사난수 생성기

5절 의사난수 생성기에 대한 공격

6절 양자난수생성기

7절 난수생성 표준: NIST SP 800-90 시리즈

제1절 난수가 사용되는 암호 기술

1.1 난수의 용도

- **키 생성**
  - 대칭 암호나 메시지 인증 코드
- **키 쌍 생성**
  - 공개 키 암호나 디지털 서명
- **초기화 벡터(IV) 생성**
  - 블록 암호 모드인 CBC, CFB, OFB
- **비표(nonce) 생성**
  - 재전송 공격 방지나 블록 암호의 CTR 모드
- **솔트 생성**
  - 패스워드를 기초로 한 암호화(PBE)
- **일회용 패드**
  - 패딩에 사용되는 열을 생성

- 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

제2절 난수의 성질

2.1 난수의 성질 분류

2.2 무작위성

2.3 예측 불가능성

2.4 재현 불가능성

2.1 난수의 성질 분류

- **무작위성**
  - randomness
  - 통계적인 편중이 없이 수열이 무작위로 되어 있는 성질

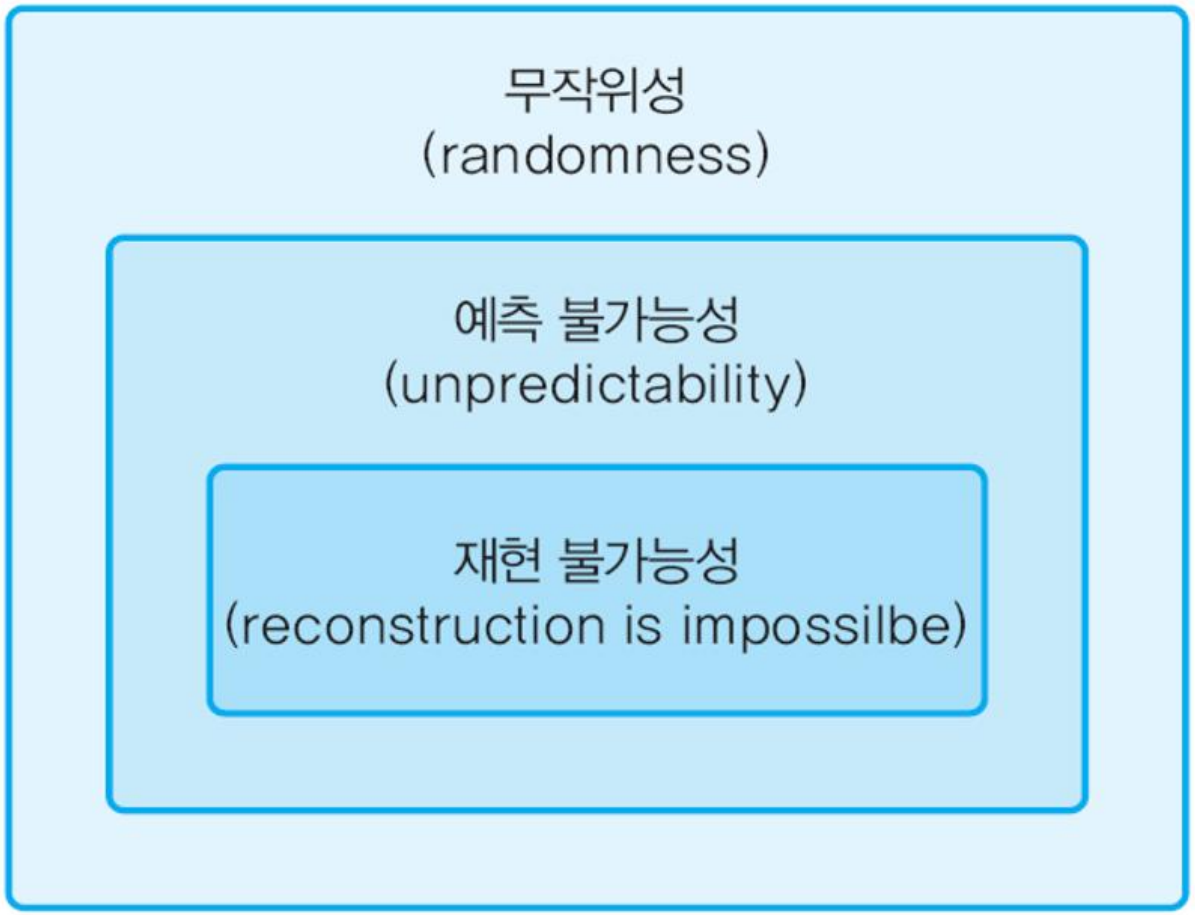
- **예측 불가능성**
  - unpredictability
  - 과거의 수열로부터 다음 수를 예측할 수 없다는 성질

- **재현 불가능성**
  - reconstruction is impossible
  - 같은 수열을 재현할 수 없다는 성질
  - 재현하기 위해서는 수열 그 자체를 보존해야만 하는 성질

난수의 분류

	무작위성	예측 불가능성	재현 불가능성	비고	암호기술적용 가능여부
약한 의사난수	○	×	×	무작위성만 갖는다	암호 기술에 사용할 수 없다
강한 의사난수	○	○	×	예측 불가능성도 갖는다	암호 기술에 사용할 수 있다
진정한 난수	○	○	○	재현 불가능성도 갖는다	

난수의 성질


무작위성
(randomness)
예측 불가능성
(unpredictability)
재현 불가능성
(reconstruction is impossilbe)

2.2 무작위성 (Randomness)

- 「아무렇게」 보이는 성질
- 의사난수열의 통계적인 성질을 조사해서 치우침이 없도록 하는 성질
  - 난수 검정
    .의사난수열의 무작위성을 조사하는 것

- 암호 기술에 사용하는 난수는 무작위성을 가지고 있는 것만으로는 불충분
  - 약한 의사난수
    .무작위성만을 갖는 의사난수

2.3 예측 불가능성 (Unprecictability)

- 공격자에게 간파당하지 않는다는 예측 불가능성이 필요
- 과거에 출력한 의사난수열이 공격자에게 알려져도 다음에 출력하는 의사난수를 공격자는 알아맞힐 수 없다는 성질
- 알고리즘은 공격자에게 알려져 있다고 가정하고 Seed를 사용
  - 시드 (seed): 공격자에게 비밀

강한 의사난수

- **약한 의사난수**
  - 무작위성만을 갖는 의사난수

- **강한 의사난수**
  - 예측 불가능성을 갖는 의사난수
  - 예측 불가능성을 가지면 당연히 무작위성을 가짐

2.4 재현 불가능성

- 한 난수열이 주어졌을때 동일한 수열을 재현할 수 없는 성질
- 재현하기 위해서는 그 난수열 자체를 보존해두는 것 이외에 방법이 없는 성질
- 소프트웨어만으로는 재현 불가능성을 갖는 난수열 생성 불가
- 소프트웨어는 의사난수열만 생성가능
  - 소프트웨어가 돌아가는 컴퓨터가 유한의 내부 상태밖에 없기 때문

주기

- 소프트웨어가 생성하는 수열은 언젠가는 반복
- 주기(period) : 반복이 다시 시작할 때 까지의 수열의 길이
- 주기를 갖는 수열은 재현 불가능하지 않음

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

재현 불가능한 난수

- 진성난수(Real Random Number):
  - 재현 불가능한 난수
  - 위의 3 가지 성질을 모두 가진다
    .무작위성
    .예측 불가능성
    .재현 불가능성
  - 예)
    . 대기 잡음: 대기에서 발생하는 자연적인 잡음 신호
    . 열 잡음: 열적 운동에 의해 발생하는 잡음 신호
    . 타 외부 전자기 및 양자 현상: 다양한 외부 환경의 물리적 현상을 측정하여 난수를 생성
    . 진공 플럭츄에이션 (Quantum Vacuum Fluctuation) - 진공상태에서도 입자/에너지의 순간적 출현
    . 양자 난수 생성기(QRNG)

3.1 의사난수 생성기의 구조

난수 생성기와 의사난수 생성기

- 난수 생성기(random number generator; RNG)
  - 하드웨어로 생성

- 의사난수 생성기(pseudo random number generator; PRNG)
  - 소프트웨어로 생성
  - 주기성을 가짐

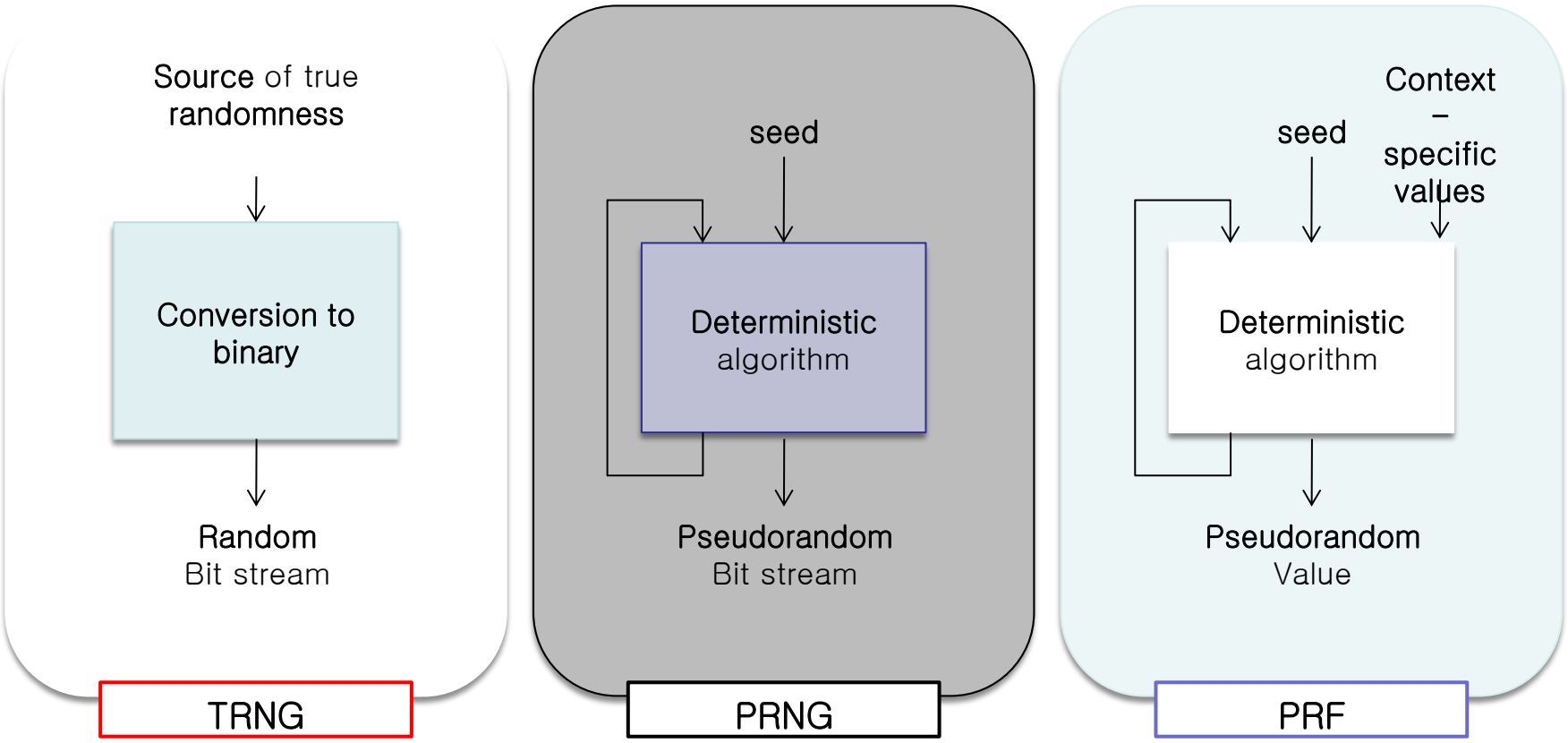
의사 난수 생성의 원리

- 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 : 의사난수 함수)
  - 고정된 길이의 의사 난수 비트열을 생산하는데 사용

- 난수와 의사난수 생성기

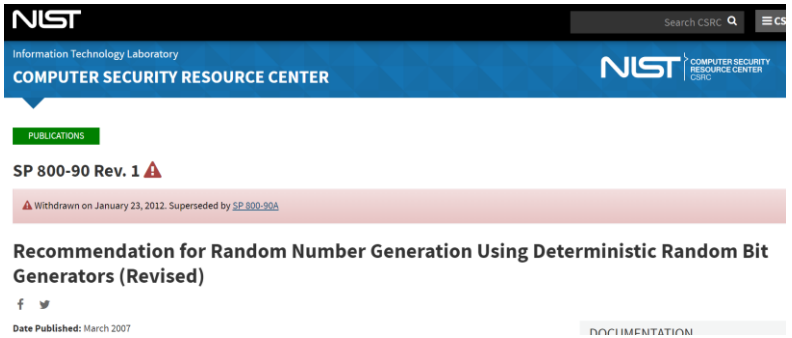

Source of true randomness
Conversion to binary
Random Bit stream
TRNG
seed
Deterministic algorithm
Pseudorandom Bit stream
PRNG
seed
Context – specific values
Deterministic algorithm
Pseudorandom Value
PRF

- PRNG 요구 사항
  - Seed를 알지못하는 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 거이 동일하게 존재
    - 확장성
      - 비트열이 랜덤하면 무작위로 추출된 어떤 비트열도 랜덤해야 함
    - 일관성
      - 생성기의 동작은 초기값 전반에 대해 일관되어야 함

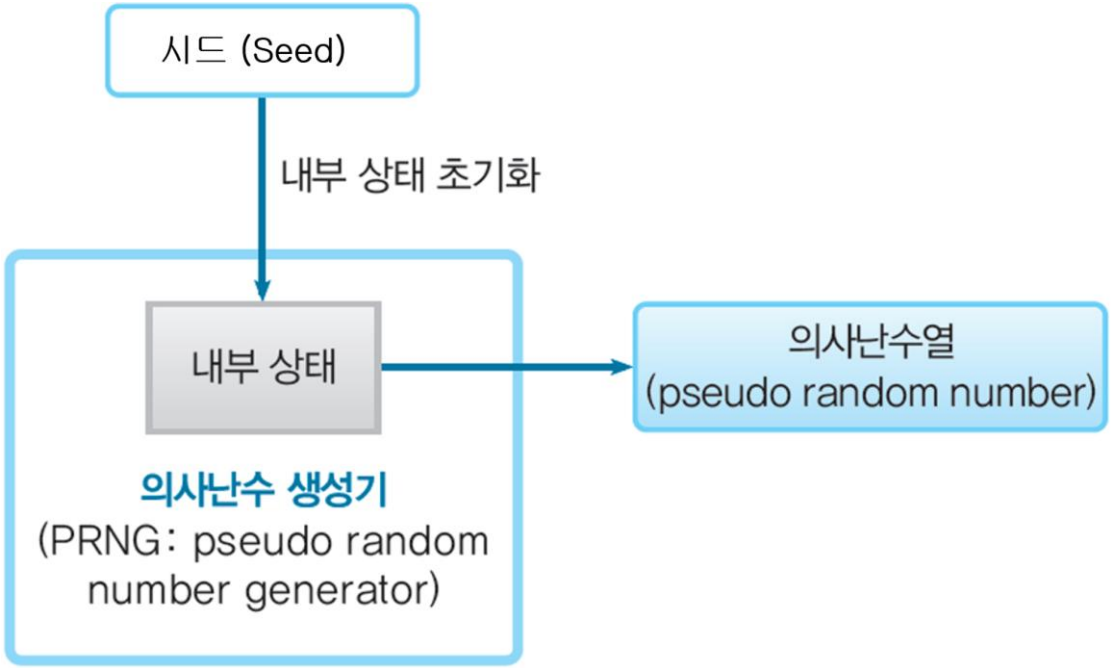
- PRNG 요구 사항(계속)
  - 비예측성(Unpredictability)
    .수열의 잇따른 다음수의 순서에 대해 예측이 불가능해야 함
    .전 방향 비예측성
      - 이전 비트들에 대한 정보가 있다고 하더라도 다음 출력 비트는 예측할 수 없어야 함
    .후 방향 비예측성
      - 생성된 어떠한 값의 정보를 통해서도 seed를 결정할 수 없어야 함

  - 시드 요구사항(Seed Requirements)
    .Seed는 예측 불가능해야 함
    .Seed는 난수 또는 의사난수이어야 함
    .TRNG에 의해 seed생성
      - SP800-90에서 권고 (NIST)



3.1 의사난수 생성기의 구조

- 내부 상태
- 시드(Seed)



- 의사난수 생성기가 관리하고 있는 메모리 값
- 의사난수 생성기는 메모리의 값(내부 상태)을 기초로 해서 계산을 수행
- 그 계산 결과를 의사난수로서 출력
- 다음 의사난수의 요구에 대비해서 자신의 내부 상태를 변화

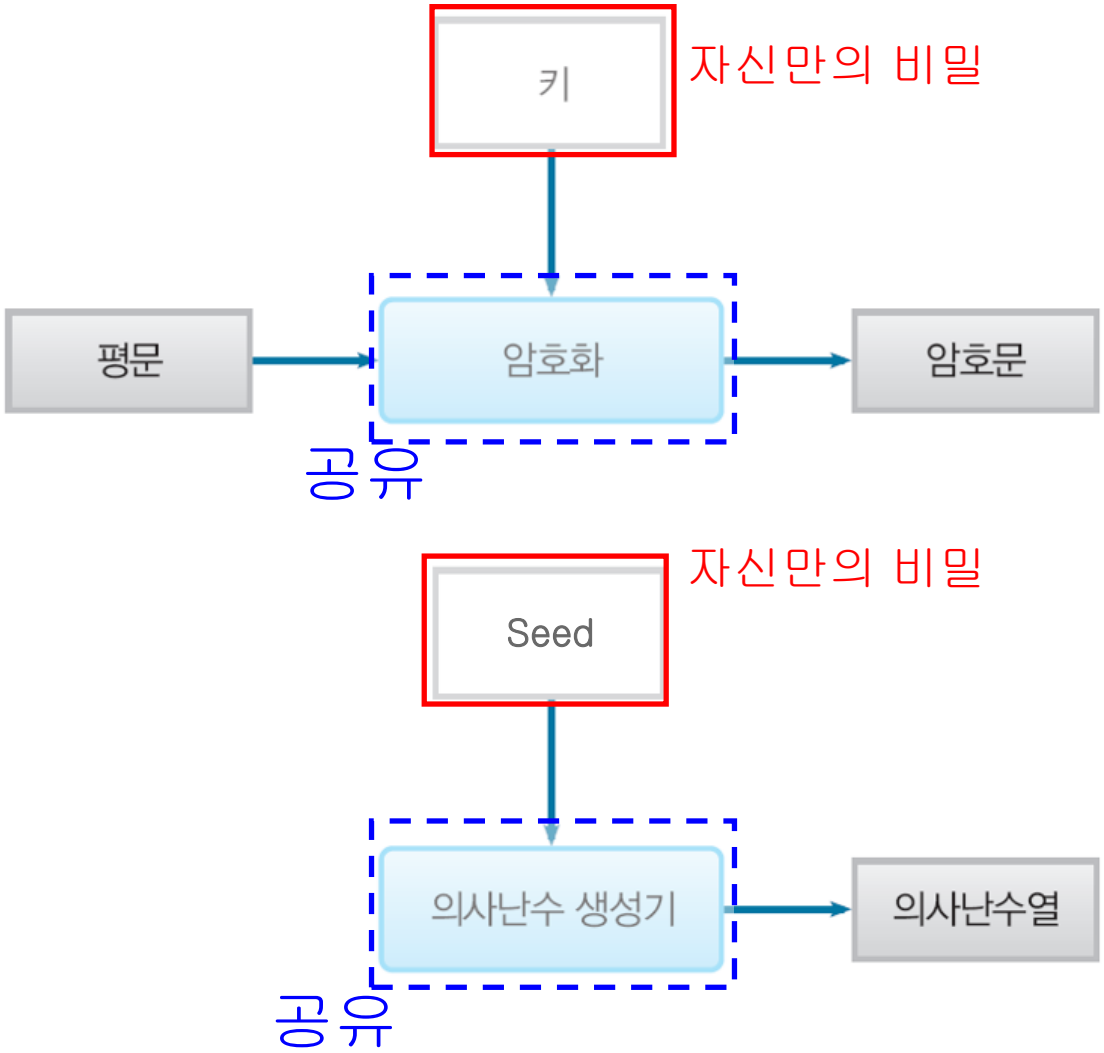
의사난수 생성 알고리즘

- 의사난수 생성 알고리즘의 2가지 기능
  - 의사난수를 계산하는 방법
  - 내부 상태를 변화시키는 방법

의사난수 생성기의 「Seed」

- 의사난수 생성기의 내부 상태 초기화에 필요
- 랜덤한 비트 열
- Seed는 자신만의 비밀로 유지

암호 키와 의사난수 Seed



제4절 구체적 의사난수 생성기

4.1 무작위 방법

4.2 선형 합동법

4.3 일방향 해시 함수를 사용하는 방법

4.4 암호를 사용하는 방법

4.5 ANSI X9.17

4.6 기타 알고리즘

4.1 무작위 방법

- **긴 주기:**
  - 암호 기술에서 사용하는 난수는 예측 불가능성을 가져야 하므로 주기가 짧아서는 안 됨

- **복잡한 알고리즘 보다는 명확한 알고리즘**
  - 프로그래머가 자세한 내용을 이해할 수 없는 알고리즘으로 생성한 난수는 예측 불가능성을 갖는지 어떤지 평가를 할 수 없음

4.2 선형 합동법

- **선형 합동법(linear congruential method)**
  - 일반적으로 가장 많이 사용되는 의사난수 생성기
  - 암호 기술에 사용하면 안됨 (“예측 불가능성”이 없음)

  - 현재 의사난수의 값을 A배하고 C를 더한 다음, M으로 나눈 나머지를 다음 의사난수로 선택

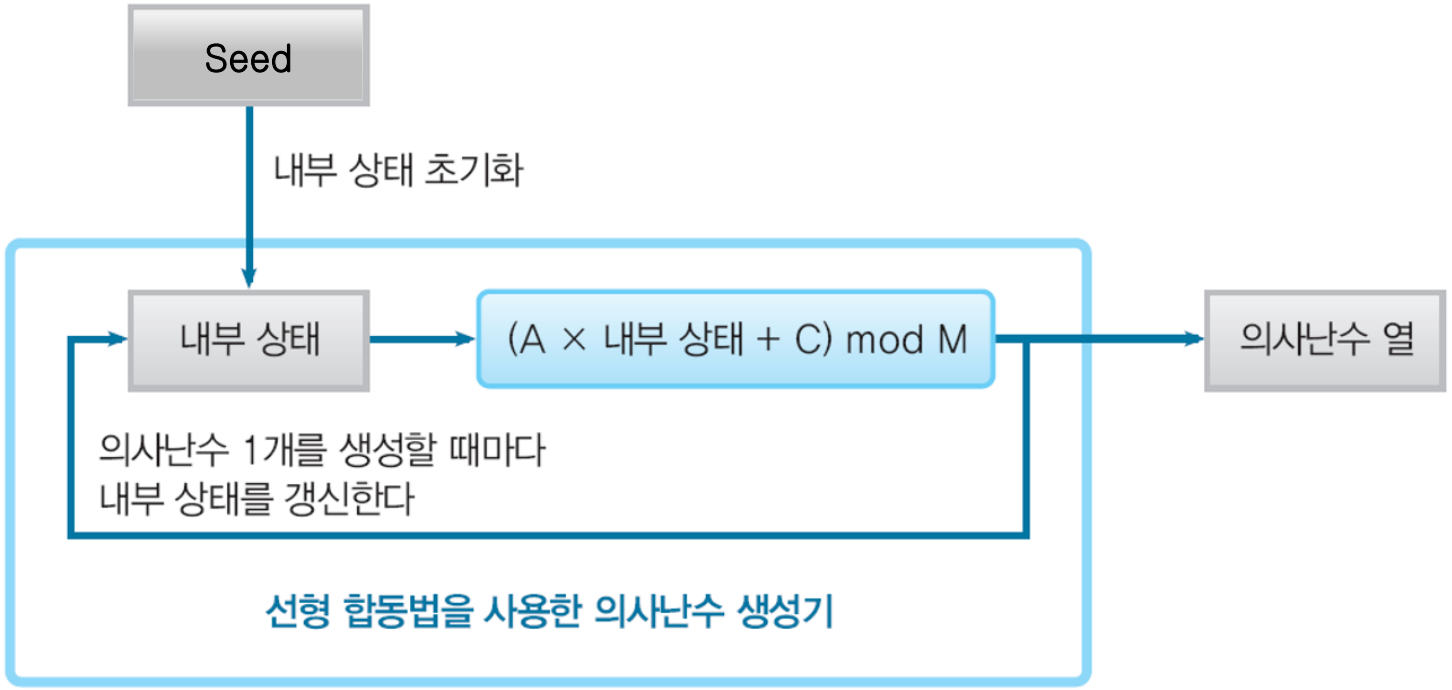
선형 합동법 계산 방법

- **R_0 생성:**
  - 최초 의사난수 $R_0 = (A \times Seed + C) \bmod M$
    - A, C, M은 정수이고, A와 C는 M보다도 작은 수

- **R_1 생성:**
  - $R_1 = (A \times R_0 + C) \bmod M$

- **R_{n+1} 생성**
  - $R_{n+1} = (A \times R_n + C) \bmod M$
    - n=2, 3…

선형 합동법에 의한 의사난수 생성기



선형 합동법 예

- A = 3
- C = 0
- M = 7

$$R_0 = (A \times \text{Seed} + C) \bmod M$$
$$= (3 \times 6 + 0) \bmod 7$$
$$= 18 \bmod 7$$
$$= 4$$

$$R_1 = (A \times R_0 + C) \bmod M$$
$$= (3 \times 4 + 0) \bmod 7$$
$$= 12 \bmod 7$$
$$= 5$$

선형 합동법 예

$$R_2 = (A \times R_1 + C) \bmod M$$
$$= (3 \times 5 + 0) \bmod 7$$
$$= 15 \bmod 7$$
$$= 1$$

$$R_3 = (A \times R_2 + C) \bmod M$$
$$= (3 \times 1 + 0) \bmod 7$$
$$= 3 \bmod 7$$
$$= 3$$

- 반복해서 4, 5, 1, 3, 2, 6, 4, 5, 1, 3, 2, 6, … 생성
- 이 경우 4, 5, 1, 3, 2, 6이라는 6개 숫자의 반복이 되므로 주기는 6

선형 합동법 다른 예

- 의사난수의 값은 M으로 나눈 나머지므로, 반드시 0부터 M-1의 범위
- A와 C와 M의 값에 따라 다른 주기를 가짐
- 예) A=6, C=0, M=7, Seed= 6인 경우
  - 의사난수열은 1, 6, 1, 6, 1, 6, …
  - 주기는 2

A값별 의사난수

- A값 변화에 따른 의사난수열

- A = 0인 경우 : 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …(주기는 1)
- A = 1인 경우 : 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, …(주기는 1)
- A = 2인 경우 : 5, 3, 6, 5, 3, 6, 5, 3, 6, 5, …(주기는 3)
- A = 3인 경우 : 4, 5, 1, 3, 2, 6, 4, 5, 1, 3, …(주기는 6)
- A = 4인 경우 : 3, 5, 6, 3, 5, 6, 3, 5, 6, 3, …(주기는 3)
- A = 5인 경우 : 2, 3, 1, 5, 4, 6, 2, 3, 1, 5, …(주기는 6)
- A = 6인 경우 : 1, 6, 1, 6, 1, 6, 1, 6, 1, 6, …(주기는 2)

선형합동법의 단점

- 선형 합동법은「예측 불가능성」이 없다
- 선형 합동법을 암호 기술에 사용해서는 절대로 안됨
  - 예: 선형 합동법을 사용하는 함수
    - C 라이브러리 함수 rand
    - Java의 java.util.Random 클래스
  - 이들을 암호 기술에서 사용해서는 안됨

선형 합동법의 예측불가능성 테스트 (공격)

- 공격자가 A=3, C=0, M=7이라는 값을 알고 있다고 가정
- 공격자가 생성된 의사난수를 1개라도 손에 넣는다면 그 다음에 생성되는 의사난수를 예측하는 것이 가능
- 입수한 의사난수 R을 이용 다음을 계산

$$(A \times R + C) \bmod M = (3 \times R + 0) \bmod 7$$

- 공격자는 의사난수의 Seed: 6을 알 필요가 없다
- 위에서 수행한 계산이 반복으로 이후 생성되는 의사난수열을 모두 예측 가능

4.3 일방향 해시 함수를 사용하는 방법

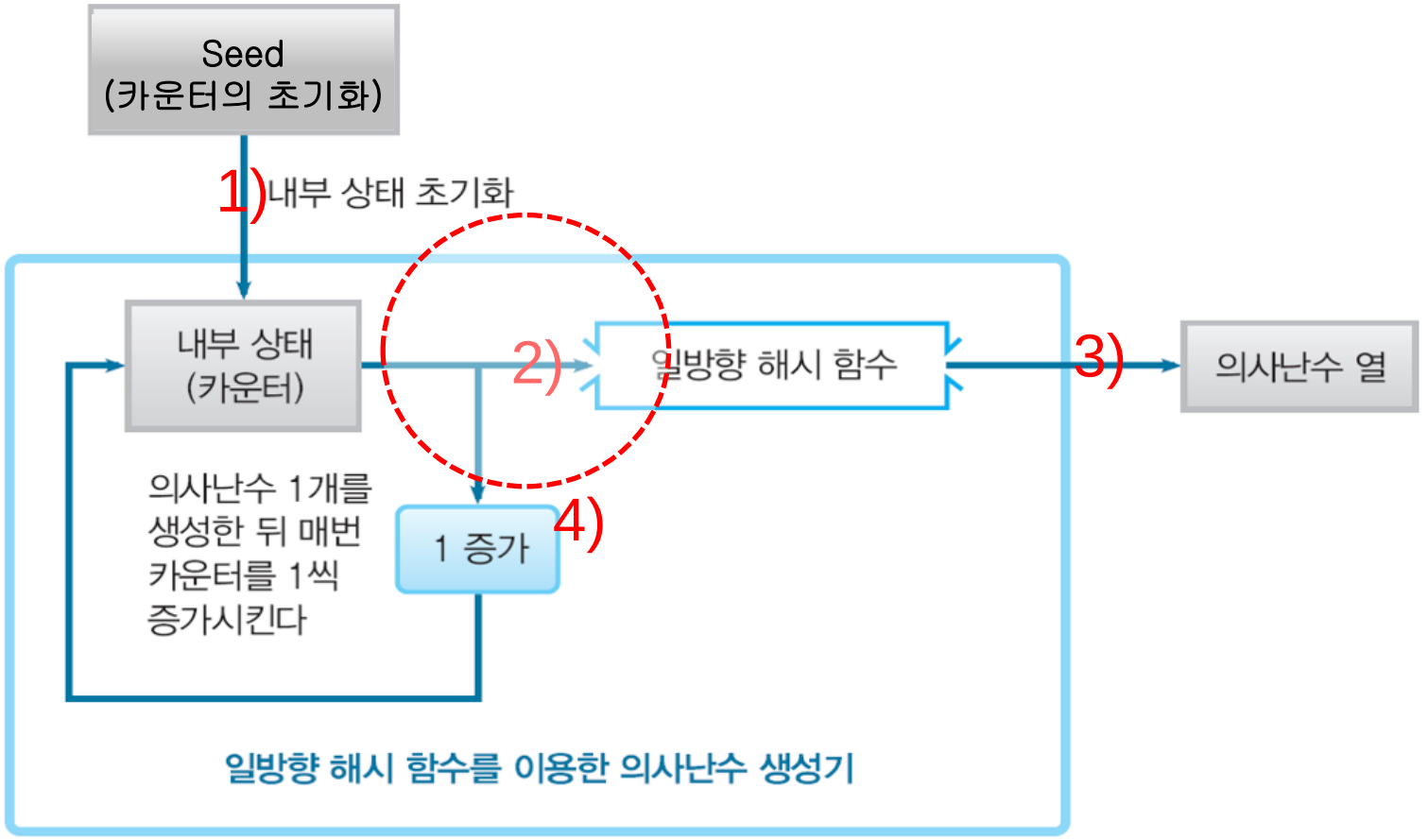
- 일방향 해시 함수(예, SHA-1)를 사용해서 예측 불가능성을 갖는 의사난수 열(강한 의사난수)을 생성하는 의사난수 생성기를 만들 수 있다

- 일방향 해시 함수의 일방향성이 의사난수 생성기의 예측 불가능성을 보장

절차

1) 의사난수의 Seed를 사용해서 내부 상태(카운터)를 초기화
2) 일방향 해시 함수를 사용해서 카운터의 해시 값 생성
3) 그 해시 값을 의사난수로서 출력
4) 카운터를 1 증가
5) 필요한 만큼의 의사난수가 얻어질 때까지 (2)~(4)를 반복

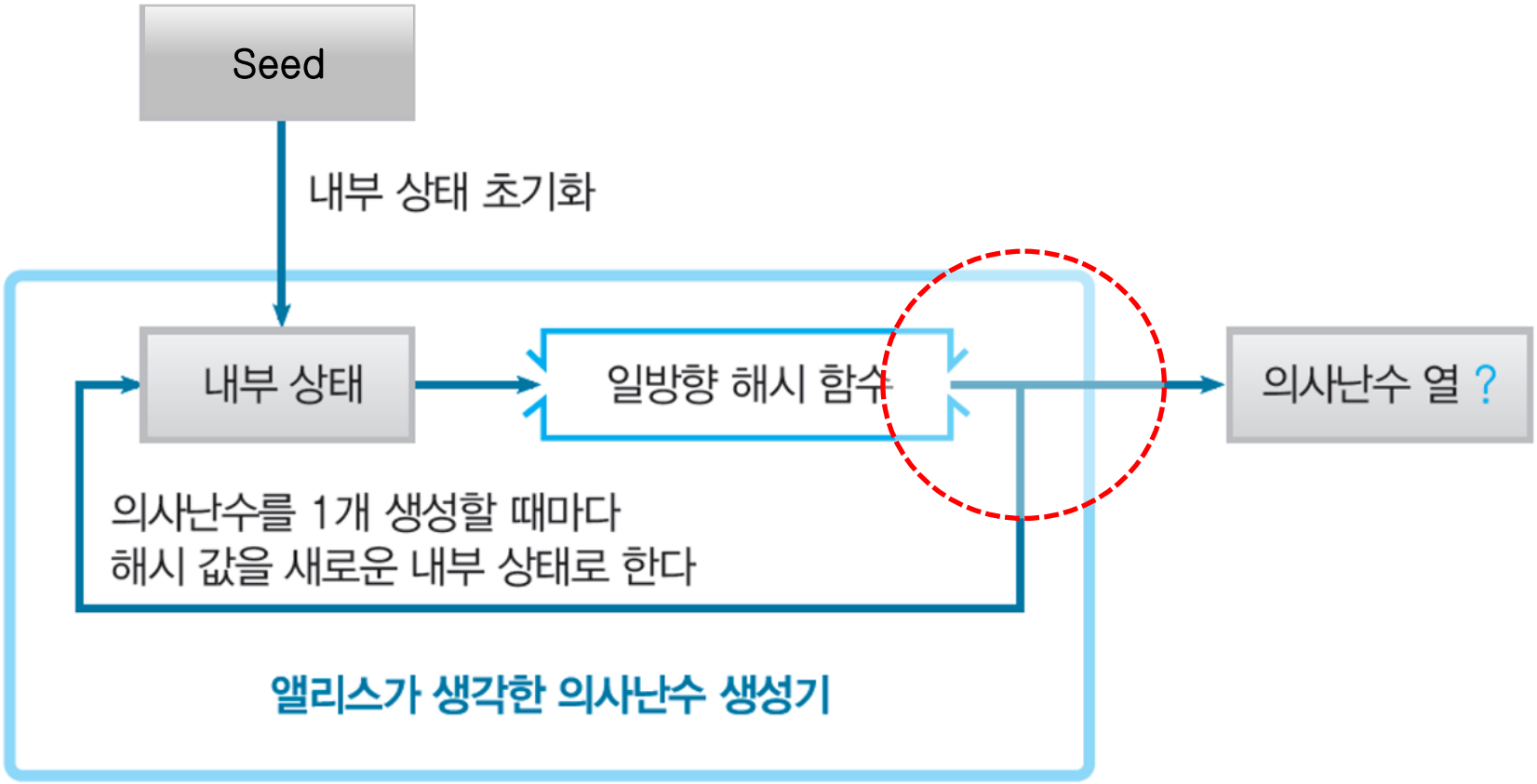
일방향 해시 함수를 사용한 의사난수 생성기



잘못 만들어진 의사난수 생성기

- 다음과 같이 생성했다고 해보자
  1) 의사난수의 Seed를 사용해서 내부 상태를 초기화한다.
  2) 일방향 해시 함수를 사용해서 내부 상태의 해시 값을 얻는다.
  3) 해시 값을 의사난수로서 출력한다.
  4) 그 해시 값을 새로운 내부 상태로 한다.
  5) 필요한 만큼의 의사난수가 얻어질 때까지 (2)~(4)를 반복한다.

잘못 만들어진 의사난수 생성기



왜 예측불가능성이 없나?

- 마지막에 출력한 의사난수의 해시 값을 취해서 다음 의사난수를 생성하므로 예측 불가능성을 갖지 않는다.
- 이유?
  - 예측 불가능성을 갖기 위해서는 일방향 해시 함수의 일방향성을 사용하는 것이 핵심
  - 출력 = 해시 입력값 ➔ 역산이 필요 없음
    .출력값이 내부 상태이므로, 해시함수의 안전성은 의미 없음

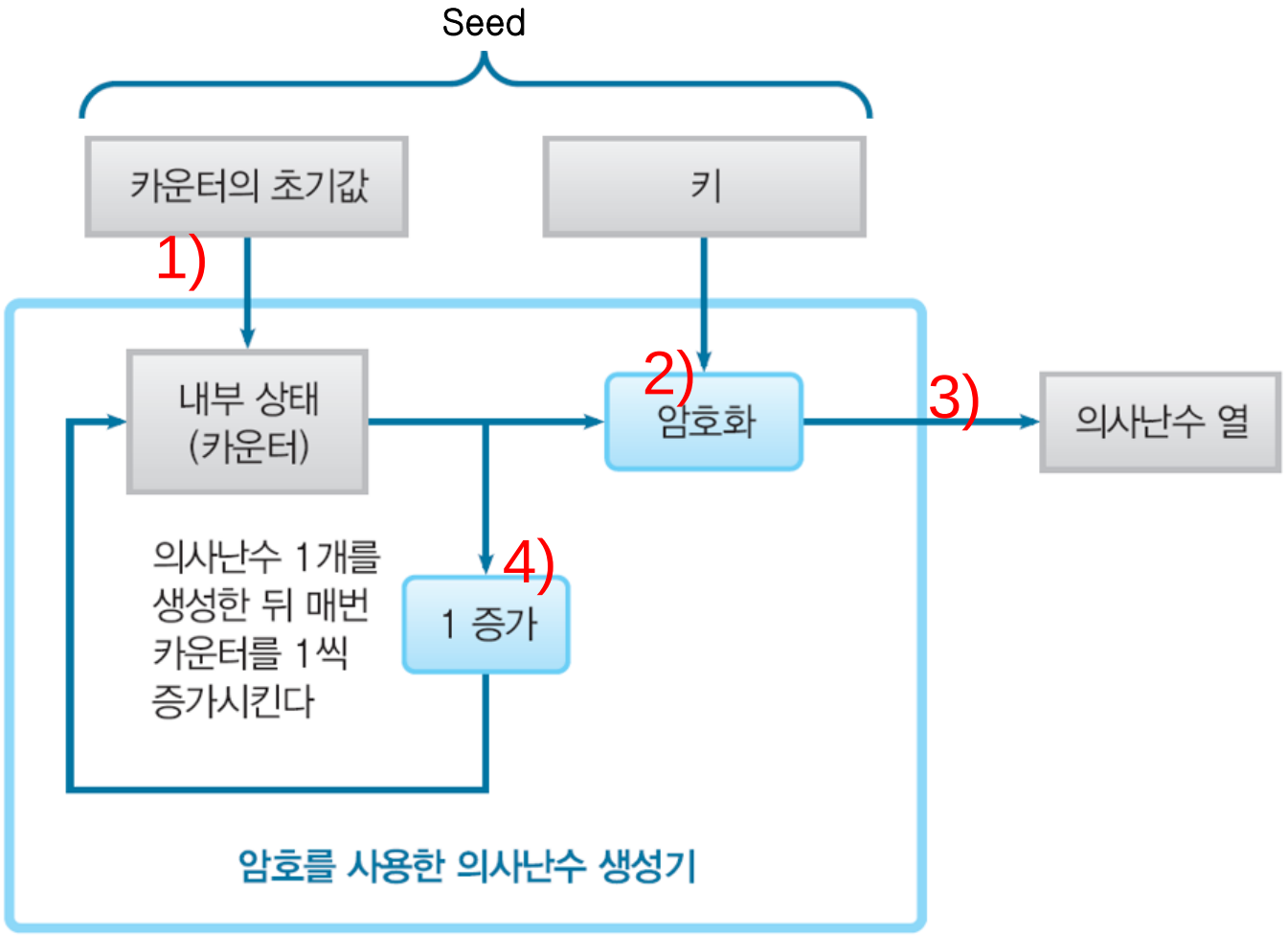
4.4 암호를 사용하는 방법

- 암호를 사용해서「강한 의사난수」를 생성하는 의사난수 생성기를 만들 수 있다
- AES와 같은 대칭 암호나 RSA와 같은 공개 키 암호 중 어느 것을 사용해도 무방
- 암호의 기밀성이 의사난수 생성기의 예측 불가능성을 보장
- 블록 암호 운용 모드를 이용한 PRNG
  - CTR모드 : SP800-90, ANSI 표준 X9.82(난수 생성), RFC4086
  - OFB모드 : X9.82, RFC4086

암호를 사용한 의사난수생성 절차

1) 내부 상태(카운터)를 초기화
2) 키를 사용해서 카운터를 암호화
3) 그 암호문을 의사난수로서 출력
4) 카운터를 1 증가
5) 필요한 만큼의 의사난수가 얻어질 때까지 (2)~(4)를 반복

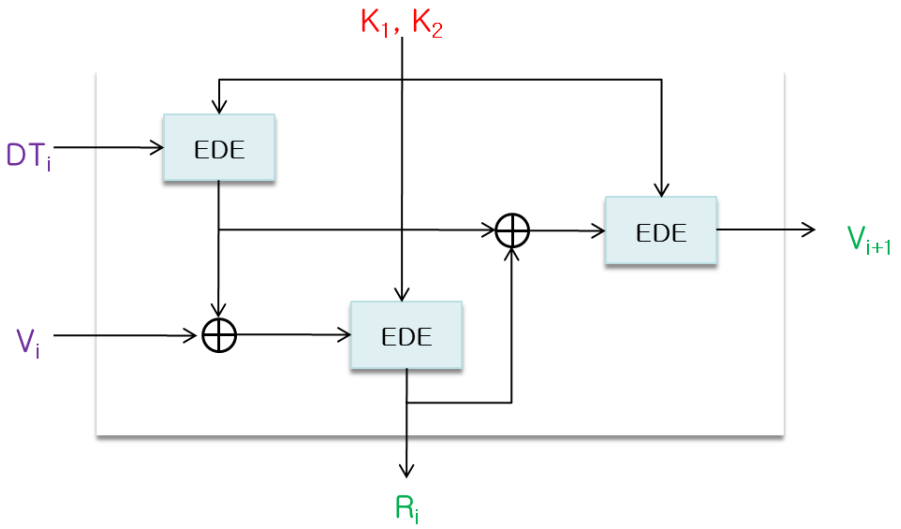
암호를 사용한 의사난수 생성기



4.5 ANSI X9.17

- 암호 소프트웨어 PGP에서 사용하는 의사난수 생성기
  - ANSI X9.17
  - ANSI X9.31

- 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



- DES의 변형인 3-DES를 사용하므로 안전성에 문제가됨

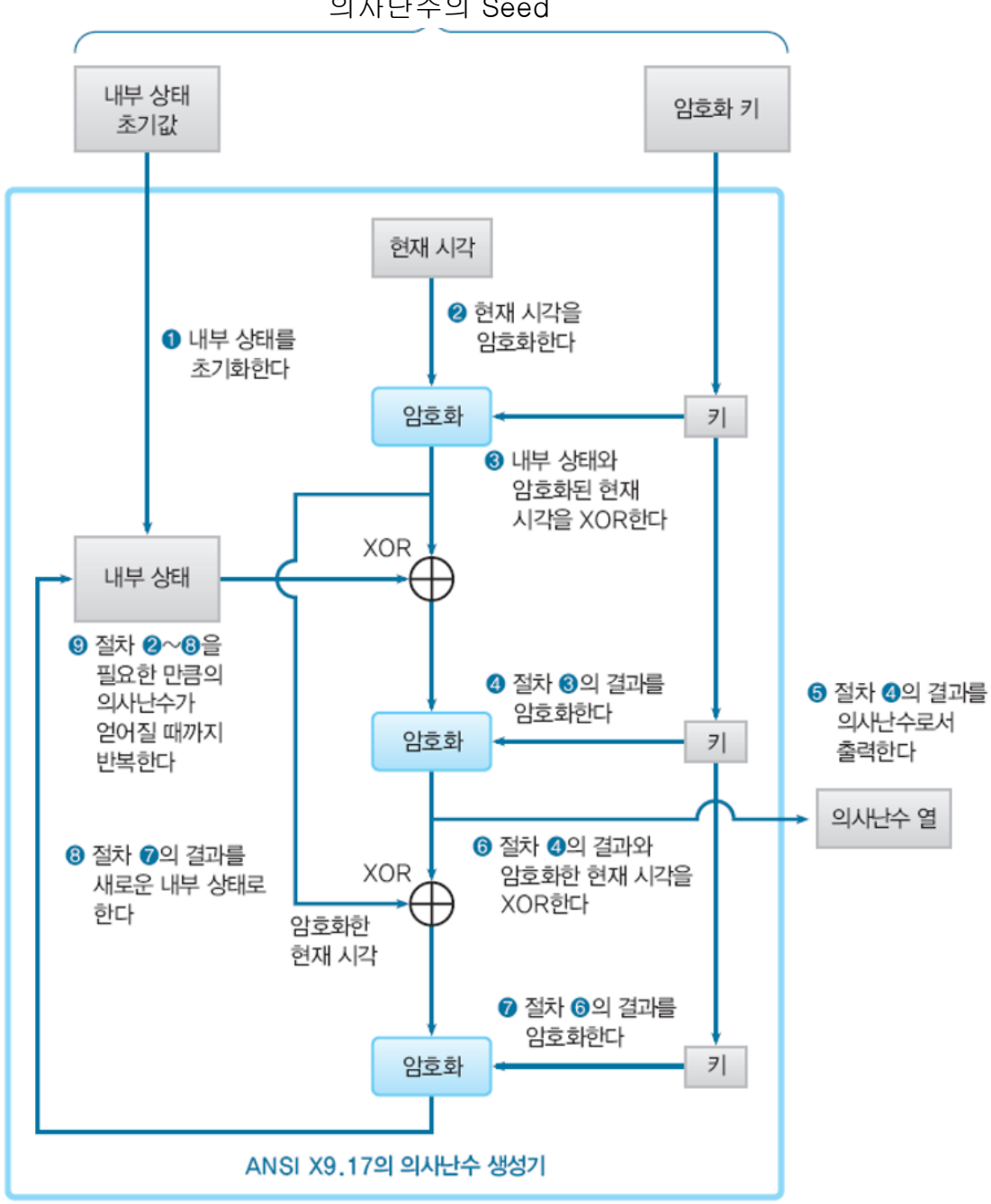
  ➔ 대안: AES를 이용한 난수 생성이 더 적합

  (NIST SP 800-90A : AES-CTR)

ANSI X9.17 의사난수 생성 절차

1) 내부 상태를 초기화
2) 현재 시각을 암호화
3) 내부 상태와 암호화된 현재 시각을 XOR
4) 절차(3)의 결과를 암호화
5) 절차(4)의 결과를 의사난수로서 출력
6) 절차(4)의 출력과 암호화된 현재시각을 XOR
7) 절차(6)의 결과를 암호화
8) 절차(7)의 결과를 새로운 내부 상태로 설정
9) 절차(2)~(8)을 필요한 만큼의 의사난수가 얻어질 때까지 반복

ANSI X9.17 방법 의사난수 생성기



왜 내부 상태 추측을 못하나?

- 공격자는 의사난수로부터 역산해서 「내부 상태와 암호화된 현재 시각의 XOR」을 간파할 수 없다
  - 이유: 간파하기 위해서는 암호를 해독해야만 한다

- 공격자는 지금까지 출력된 의사난수열로부터 의사난수 생성기의 내부 상태를 추측 못한다

4.6 기타 알고리즘

- **알고리즘을 선택할 때 주의 점**
  - 반드시「이 난수 알고리즘은 암호나 보안 용도로 사용할 수 있는가」를 확인
  - 난수로 뛰어난 알고리즘이라 하더라도 예측불가능성을 갖추지 못한 것은 암호나 보안 용도로는 사용하면 안 됨
    . 메르센 트위스터(Mersenne Twister: MT)
      - 유명한 의사 난수 생성 알고리즘이지만, 보안용으로는 사용하면 안 됨
      - Python의 random 모듈은 기본적으로 MT알고리즘을 사용 (비추천)

    . 선형 합동법
      - 충분한 길이의 난수열을 관찰하면 앞으로 생성될 난수열이 예측가능하므로 사용하면 안 됨
  - 보안상 안전성이 깨진 암호나 해시함수를 사용하면 안됨
    . DES, MD4, MD5 등

안전한 알고리즘

- JAVA – java.security.SecureRandom 클래스
  - Java.util.Random – 보안 취약, 사용금지
- Ruby – SecureRandom 모듈 사용
  - Random 클래스 – 보안 취약, 사용금지

- C – getrandom(), RAND_bytes(), BCryptGenRandom()
  - rand(). Srand(),random() : 사용하지 말것
- Python – secrets.token_bytes(n), secrets.randbelow(n)
- JavaScript – crypto.randomBytes(n)
- .NET RNGCryptoServiceProvider, andomNumberGenerator.GetBytes()

제5절 의사난수 생성기에 대한 공격

5.1 Seed에 대한 공격

5.2 랜덤 풀에 대한 공격

5.3 기타 공격

5.1 Seed에 대한 공격

- **Seed의 중요성**
  - 의사난수의「Seed」는 암호의「키」에 필적
  - Seed가 공격자에게 노출 되면, 그 의사난수 생성기가 생성한 모든 의사난수열은 공격자에게 노출

- **Seed 선택**
  - 재현 불가능성을 갖는 「진정한 난수」 선택

5.2 랜덤 풀에 대한 공격

- **랜덤 풀**
  - Seed로 사용할 랜덤 비트 열을 사전에 만들어 비축해 놓은 파일
  - 암호 소프트웨어가 의사난수 Seed가 필요할 경우 필요한 만큼의 랜덤한 비트 열을 꺼내서 사용
  - 랜덤 풀 자체는 특별한 정보를 가지지 않지만 유익한 정보 저장을 위해 필요하므로 잘 지켜야 함

5.2 기타 공격

- **선형성 공격 (Linearity Attack):** 일부 PRNG가 선형적인 관계를 가지므로, 초기값이나 중간 값을 알아 유추
- **주기성 분석 공격 (Cycle Detection Attack)** : PRNG의 유한한 상태 공간에서 주기를 찾아 반복성을 분석하여 유추
- **통계적 분석 (Statistical Attack):** PRNG의 출력이 통계적으로 예측하거나 편향을 찾아내는 공격
- **시계 기반 공격 (Time-based attack):** 시스템 시간이 시드로 사용될 경우, 가능한 시드 값을 제한하여 공격자가 시드를 유추
- **사이드 채널 공격 (Side-Channel Attack)** :PRNG 작동 중 전력 소비, 시간 차이, 캐시 접근 패턴 등을 통해 내부 상태를 유추

제6절 양자난수생성기 (QRNG)

- 양자난수생성기 – QRNG (Quantum Random Number Generator)
- 양자 특성을 이용해 패턴 분석 자체가 불가능한 무작위 숫자를 만드는 장치

- 양자암호통신의 핵심 기술 (난수의 조건)
  - 예측불가능(Unpredictable)
  - 어느 한쪽으로 편향성이 없어야 하고(Unbiased)
  - 앞뒤 숫자(bit stream)간에 상호연관성이 없어야(Uncorrelated)함

- 특징: 양자난수생성기로 만든 난수는 패턴 연산 컴퓨팅으로도 풀 수 없는 불규칙성을 가지고 있어 보안성이 높음

- QRNG는 기존 보안체계의 한계를 극복할 수 있는 차세대 핵심 기술

- QRNG 기술은 양자 암호, 양자 컴퓨팅, 양자 통신 등 차세대 보안 시스템 구축에 필수 요소로 부상 중

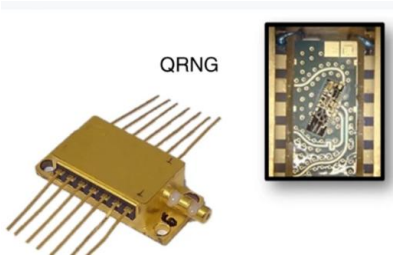
- IDQ의 QRNG 사용 사례

삼성 갤럭시 퀀텀 5 스마트폰

QRNG로 운영하는 EzQuant 보안키

DocuSign, 양자 보안 전자 서명

- QRNG Chip / Board



6.1 QRNG 국내동향

- 한국표준과학연구원(KRISS): 단일광자 큐비트 상태 평가 기술 개발, QRNG 관련 표준화 활동 수행
- 한국전자통신연구원(ETRI): 양자 난수 생성기, 양자 통신 시스템, 단일광자 검출기 등 핵심 기술 연구 및 산업화 추진
- SK텔레콤: QRNG 기반 보안칩 'Q-HSM', 삼성전자와 협업한 양자보안 스마트폰(갤럭시 퀀텀 시리즈) 출시
- KT: 양자암호통신 기술과 QKD 시범 구축, 10km 구간 QKD 시연
- LG유플러스: PQC 기반 클라우드 보안 플랫폼 'Alphakey' 개발, QENC 양자암호장비 시연
- 우리로, EYL: QRNG 관련 스타트업 기술 및 장비 개발, 투자 유치 중

6.2 QRNG 해외동향

- 미국 (NIST): PQC와 QRNG 포함 차세대 암호화 기술 연구, 양자 컴퓨터에도 안전한 암호 알고리즘 개발
- 유럽연합(EU): 양자 컴퓨팅 및 양자 통신 기술 개발 위한 대규모 프로젝트 지원, 양자 암호화 기술 투자 확대
- 중국: 양자위성 발사 및 독립적 연구 생태계 조성, 국제 경쟁 주도권 확보 시도
- 일본: 300kbps×45km 양자암호 장치 개발 및 제품화 성공, QRNG 등 관련 기술 상용화 진행

제7절 난수생성 표준: NIST SP 800-90

- 발행기관: 미국 국립표준기술원(NIST)

- 목적: 안전하고 예측 불가능한 난수 생성을 위한 표준 제공

- 시리즈 구성
  - SP 800-90A: 암호학적 난수 생성기(DRBG) 설계 및 운용
  - SP 800-90B: 엔트로피 소스의 보안성 평가 및 요구사항
  - SP 800-90C: 전체 RNG 시스템 아키텍처 설계 지침

- 활용
  암호 프로토콜 설계, 하드웨어 보안 모듈(HSM), IoT 보안칩 등

7.1 SP 800-90A Rev.1

- DRBGs (Deterministic Random Bit Generators)
  - 발행: 2015 (Final)
- 입력된 시드(seed)와 내부 상태(state)를 기반으로 반복적으로 난수를 생성
- 완전히 예측 불가능한 난수를 생성하는 것은 아니지만, 충분한 보안성을 제공
  → 안전한 의사난수

- 지원 알고리즘
  - Hash_DRBG (SHA-2 기반 해시 함수)
  - HMAC_DRBG (HMAC-SHA2 기반)
  - CTR_DRBG (AES 대칭키 블록암호 기반)

- 중요 요구사항
  - 시드 생성: 엔트로피 소스로부터 초기 시드 획득
  - 재시딩(reseding): 일정 주기마다 엔트로피로 상태 재초기화
  - 내부 상태 보호: 상태 노출 방지 필수

- 보안 고려사항
  - 고장 대응 메커니즘
  - 시드 값 보호 및 예측 가능성 제거

7.2 SP 800-90B

- Entropy Sources
  - 발행: 2018 (Final)
- 진정한 난수(진짜 난수)를 제공하는 엔트로피 소스 평가 및 요구사항 정의

- **구성 요소**
  - Noise Source: 물리적 현상 기반 (열잡음 등)
  - Digitizer: 아날로그 신호를 디지털로 변환
  - Conditioner: 바이어스 제거 및 정규화

- **평가 요소**
  - 최소 엔트로피(min-entropy) 측정
  - Online Health Tests: 지속적 이상 탐지
    - Repetition Count Test / Adaptive Proportion Test

- **보안성 확보 조건**
  - 엔트로피 추정 신뢰성 확보
  - 무결성 보장 및 외부 영향 최소화

7.3 SP 800-90C

- Random Bit Generator (RBG) Architectures
  - 발표: 2023.9월 (Draft-초안)
- DRBG (90A)와 Entropy Sources (90B)를 조합하여 강건한 RBG 구성 방식 정의

- **구성 방식**
  - Full Entropy Mode
    - 90B 기준을 만족하는 엔트로피 소스만으로 난수 생성
    - 고보안 환경에 적합
  - Mixed Mode
    - 엔트로피 소스 + DRBG 조합
    - 고속성과 예측불가능성 보장

- **설계 고려사항:**
  - 엔트로피 공급 주기 설정
  - 시드 관리 및 재생성 조건
  - 시스템 전체 상태 관리 및 통합 테스트

13장 난수

연습문제 풀이

연습문제

1. 다음 중 난수가 사용되는 곳이 아닌 예는?

① 키 생성

② 초기화 벡터 생성

③ 비표(nonce) 생성

④ 솔트(salt) 생성

⑤ 답 없음

연습문제

2. 난수를 사용하는 주요한 목적으로 적합한 것은?

① 암호문을 길게 만들기 위해서

② 예측불가능성을 부가하여 간파당하지 않기 위해서

③ 짧은 평문을 길게 만들기 위해서

④ 복호화 과정을 단순화 하기 위해서

⑤ 초기화 벡터의 사용을 피하기 위해서

연습문제

3. 난수를 분류하는 데 사용되는 항목이 아닌 것은?

① 무작위성

② 예측 불가능성

③ 재현 불가능성

④ 주기성

⑤ 답 없음

연습문제

4. 난수 생성기와 의사난수 생성기의 차이점을 말한 것으로서 맞는 것은?

① 의사난수 생성기로 난수를 생성하는 것이 더 쉽다.

② 의사난수 생성기에서 생성된 난수는 재현이 불가능하다.

③ 의사난수 생성기가 비용이 많이 든다.

④ 난수 생성기로 생성된 난수의 주기가 의사난수 생성기로 생성한 난수의 주기보다 길다.

⑤ 난수 생성기는 하드웨어적으로 구성된다.

연습문제

5. 다음중 의사난수를 생성하는 방법이 아닌 것은?

① SHA-1을 이용하는 방법

② 선형 합동법

③ AES를 사용하는 방법

④ X.509

⑤ ANSI X9.17

연습문제

6. 선형 합동법으로 생성하는 난수는 난수의 성질 중에서 _____만 갖는다.

① 예측 불가능성

② 주기성

③ 무작위성

④ 재현 불가능성

⑤ 반복성

연습문제

7. 의사난수 생성기로 의사난수를 생성하기 위해서는 _____(이)라고 하는 정보가 필요하다. 의사난수의 _____(은)는 의사난수 생성기의 내부 상태를 초기화하는데 이용된다.

① Seed

② MAC

③ 서명

④ 패딩

⑤ 해시 값

연습문제

8. PRNG는 무엇을 나타내는 말인가?

① 디지털 서명기

② 의사난수 생성기

③ 난수 생성기

④ 암호 알고리즘

⑤ 해시 함수

연습문제

9. 과거의 수열로부터 다음 수를 알 수 없다는 성질을 무슨 성질이라고 하는가?

① 미래 불규칙성

② 예측 불가능성

③ 재생 불가능성

④ 무작위성

⑤ 반복 불가능성

참고문헌

- 암호학과 네트워크 보안, Behrouz A. Forouzan 지음, 이재광외 3인 역, 한티미디어
- 컴퓨터 보안과 암호, WILLIAM STALLINGS 지음, 최용락외 2인 역, 그린출판사
- 네트워크 보안 에센셜 3판, 윌리엄 스톨링스 저(전태일 등역) , 교보문고
- Ferguson, N., et al., Cryptography Engineering: Design Principles and Practical Applications, Wiley
- Kelsey, J., et al., Cryptanalytic Attacks on Pseudorandom Number Generators. In Fast Software Encryption (FSE) '98, Springer
- L'Ecuyer, P., Random Numbers for Simulation., Communications of the ACM, Volume 33, Issue 10
- QRNG 사용사례, IDQ / 도시바 / EIPEKS
- 양자난수생성기, SKT Insight
- Quantum Cryptography Initiatives, NIST
- 양자통신 보고서, 한국전자통신연구원(ETRI)
- 양자 연구자료, 한국표준과학연구원(KRISS)
- SKT철통보안 5G 양자기술 대중화 첫발, 전자신문
- Recommendation for Random Number Generation Using Deterministic Random Bit Generators , NIST SP 800-90A Revision 1
- Recommendation for the Entropy Sources Used for Random Bit Generation, NIST SP 800-90B
- Recommendation for Random Bit Generator (RBG) Constructions, NIST SP 800-90C (Draft)

Q & A

Thanks!