알기 쉬운

정보보호개론 3판

흥미로운 암호 기술의 세계

INFORMATION SECURITY and CRYPTOGRAPHY

INFINITYBOOKS
인피니티북스

INFORMATION SECURITY and CRYPTOGRAPHY

CHAPTER 13 난수

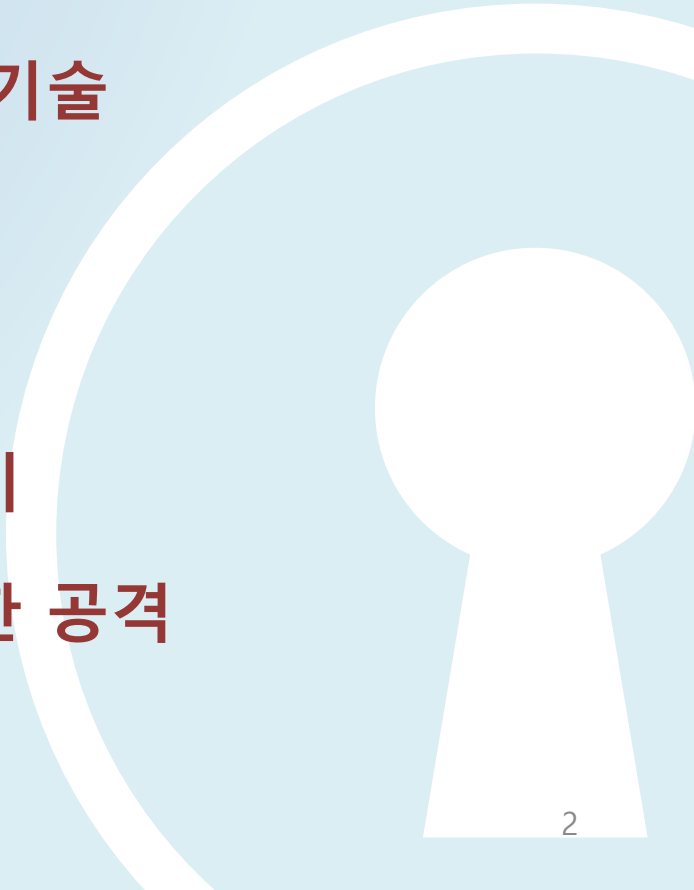
01: 난수가 사용되는 암호 기술

02: 난수의 성질

03: 의사난수 생성기

04: 구체적 의사난수 생성기

05: 의사난수 생성기에 대한 공격



Section 01
난수가 사용되는 암호 기술

1.1 난수의 용도

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

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 목적이 **간파당하지 않도록 하기 위한** 것

Section 02
난수의 성질

2.1 난수의 성질 분류

2.2 무작위성

2.3 예측 불가능성

2.4 재현 불가능성

2.1 난수의 성질 분류

- **무작위성**
  - 통계적인 편중이 없이 수열이 무작위로 되어 있는 성질
- **예측 불가능성**
  - 과거의 수열로부터 다음 수를 예측할 수 없다는 성질
- **재현 불가능성**
  - 같은 수열을 재현할 수 없다는 성질. 재현하기 위해서는 수열 그 자체를 보존해야만 하는 성질

난수의 분류

표 13-1 난수의 분류

	무작위성	예측 불가능성	재현 불가능성		
약한 의사난수	○	×	×	무작위성만 갖는다	암호 기술에 사용할 수 없다
강한 의사난수	○	○	×	예측 불가능성도 갖는다	암호 기술에 사용할 수 있다
진성 난수	○	○	○	재현 불가능성도 갖는다	

난수의 성질

그림 13-1 • 난수의 성질

2.2 무작위성

- **무작위성(Randomness)**
  - 「아무렇게」로 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

- **예측 불가능성(Unpredictability)**
  - 공격자에게 간파당하지 않는 성질
  - 과거에 출력한 의사난수열이 공격자에게 알려져도 다음에 출력하는 의사난수를 공격자는 알아맞힐 수 없다는 성질
  - 알고리즘은 공격자에게 알려져 있다고 가정하고 종자를 사용
    - **종자(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

- 재현 불가능성
  - 한 난수열이 주어졌을 때 동일한 수열을 재현할 수 없는 성질
  - 재현하기 위해서는 그 난수열 자체를 보존해 두는 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
- 반복이 다시 시작할 때까지의 수열의 길이를 **주기(period)**라고 함
- 주기를 갖는 수열은 재현 불가능하지 않음

재현 불가능한 난수 생성

- 재현 불가능한 물리 현상으로부터 정보를 취득
  - 예
    - 주위의 온도나 소리의 변화
    - 사용자의 마우스 위치 정보
    - 키 스트록 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: 동전 던지기 결과로 얻어지는 비트
  - 앞: 0
  - 뒤: 1

Quiz 1 주사위

- 주사위를 반복해서 던져서 만드는 수열은 재현 불가능성을 갖는다고 생각되는가?

칼럼 RDSEED RDRAND

- 인텔사의 새로운 CPU에서 디지털 난수 생성기가 내장되어 있고 RDSEED와 RDRAND라는 난수생성 명령어를 사용할 수 있다. 이 CPU에서는 난수를 만들어 내는 기틀 즉, 엔트로피 소스로서 회로 내부의 열 잡음이라고 하는 자연현상을 이용한다. 그리고 엔트로피 소스에서 얻어진 재현 불가능성을 가진 비트열을 AES-CBC-MAC에 걸어 콘디션드 엔트로피 샘플이라는 256비트 열을 만든다. AES-CBC-MAC라고 하는 것은 블록 암호 AES를 CBC모드에서 사용한 메시지 인증코드로 여기에서는 긴 비트열을 256비트로 정리하는 역할을 하고 있다.

Section 03
의사난수 생성기

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

3.1 의사난수 생성기의 구조

- 내부상태
- 종자(seed)

의사난수 생성기의 구조

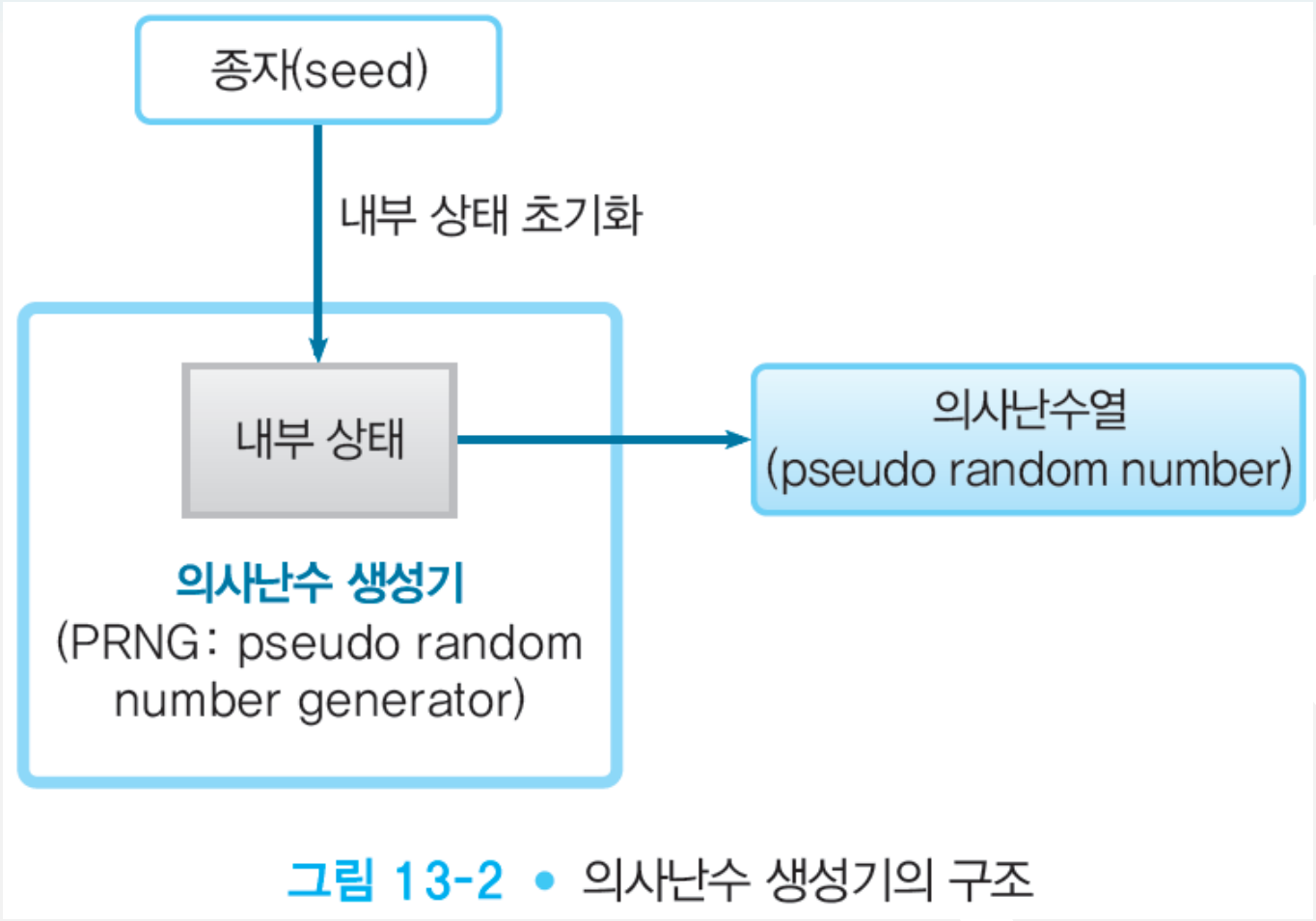


그림 13-2 • 의사난수 생성기의 구조

의사난수 생성기의 내부 상태

- 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 알고리즘은 다음 2 가지 기능을 합한 것
  - 의사난수를 계산하는 방법
  - 내부 상태를 변화시키는 방법

의사난수 생성기의「종자」

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

암호 키와 의사난수 종자

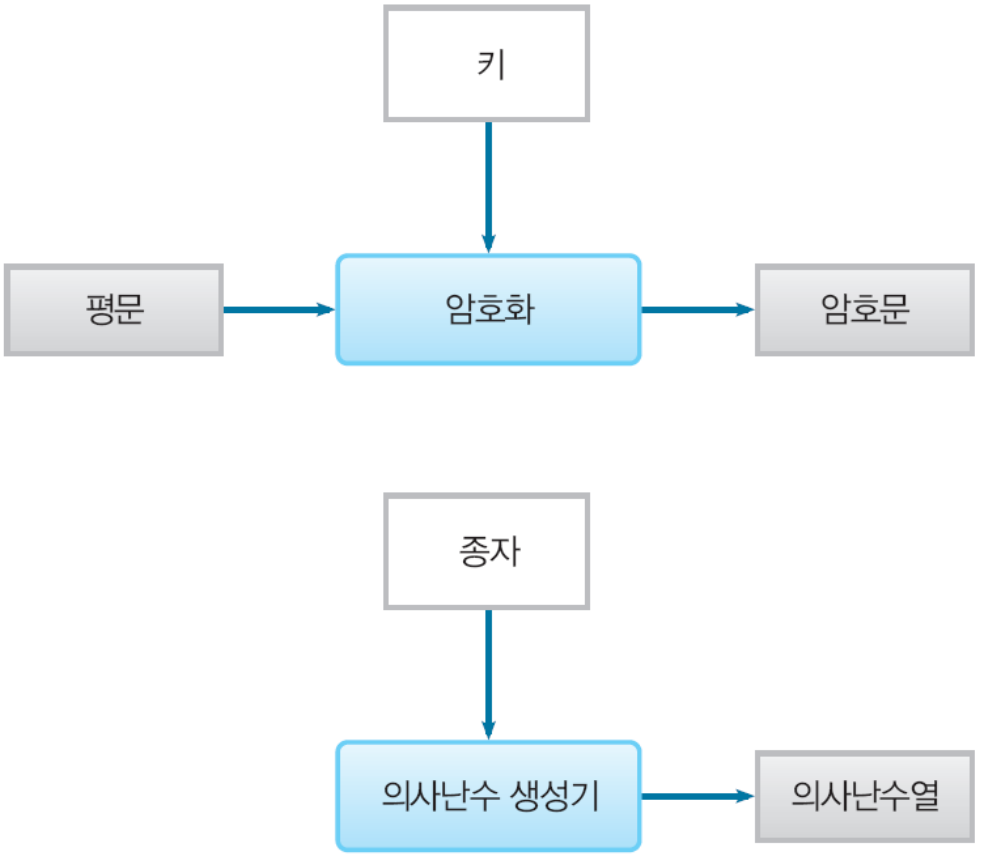


그림 13-3 • 암호 키와 의사난수의 종자

Section 04
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
  - 현재 의사난수의 값을 A배하고 C를 더한 다음, M으로 나눈 나머지를 다음 의사난수로 선택

선형 합동법 계산방법

- R_0 생성:
  - 최초 의사난수 $R_0 = (A \times 종자 + C) \bmod M$
    - 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

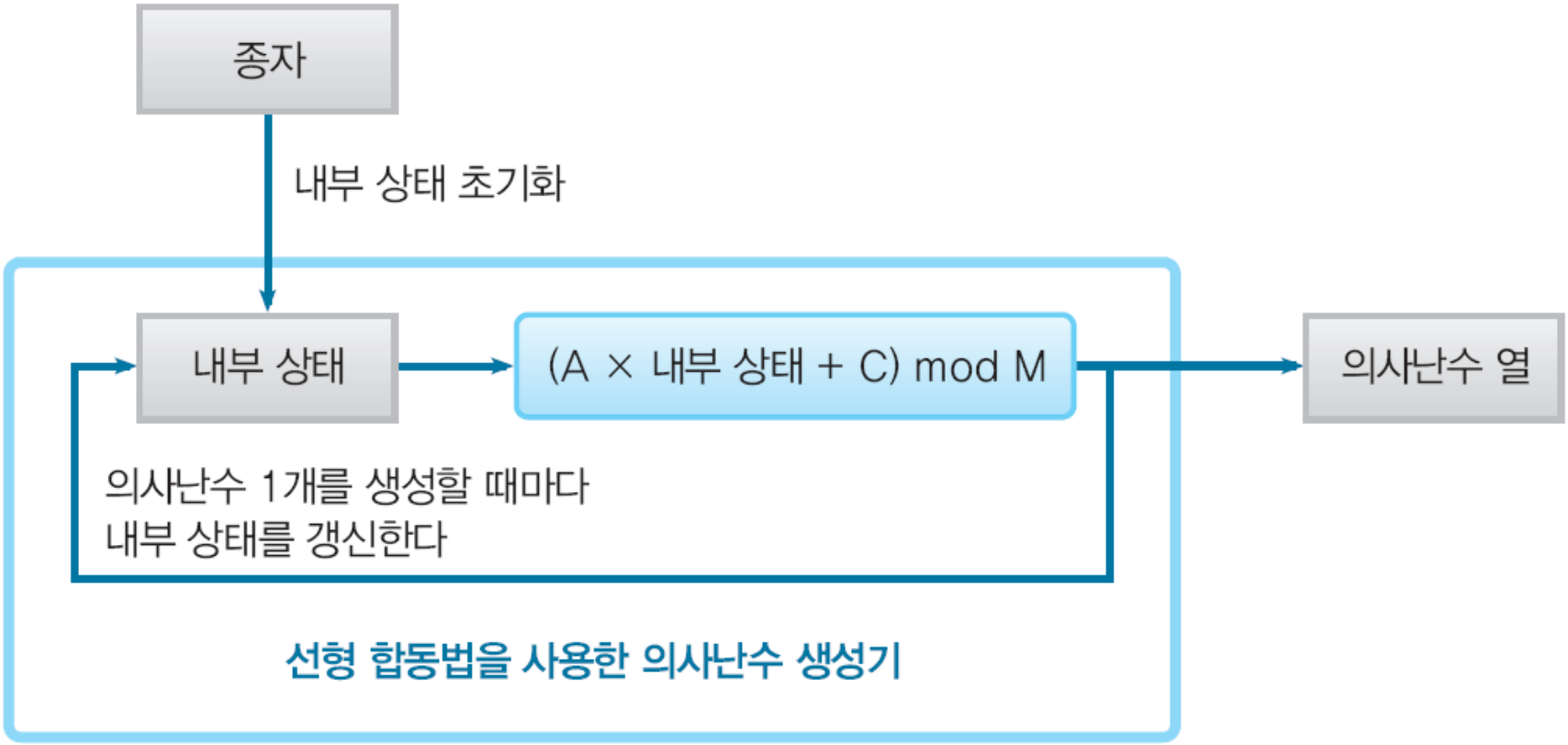


그림 13-4 • 선형 합동법에 의한 의사난수 생성기

선형합동법 예

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

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

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

선형합동법 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

선형합동법 다른 예

- 의사난수의 값은 M으로 나눈 나머지이므로, 반드시 0부터 M-1의 범위
- A와 C와 M의 값에 따라 다른 주기를 가짐
- 예: A=6, C=0, M=7, 종자= 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: 선형합동법을 사용하는 함수
    - C 라이브러리 함수 rand
    - Java의 java.util.Random 클래스
  - 이들을 암호 기술에서 사용해서는 안됨

선형합동법의 예측불가능성 테스트

- 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 × R + C) mod M = (3 × R + 0) mod 7

칼럼 선형합동법 프로그램 개요

M=양의 정수;

A=M보다 작은 0 이상의 정수;

C=M보다 작은 0 이상의 정수;

내부 상태 = 의사난수 종자;

```
while(true) {
    의사난수=(A x 내부 상태 + C) mod M;
    내부상태=의사난수;
    의사난수를 출력한다.
    }
```

4.3 일방향 해시 함수를 사용하는 방법

- 일방향 해시 함수(예를 들면 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 종자를 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

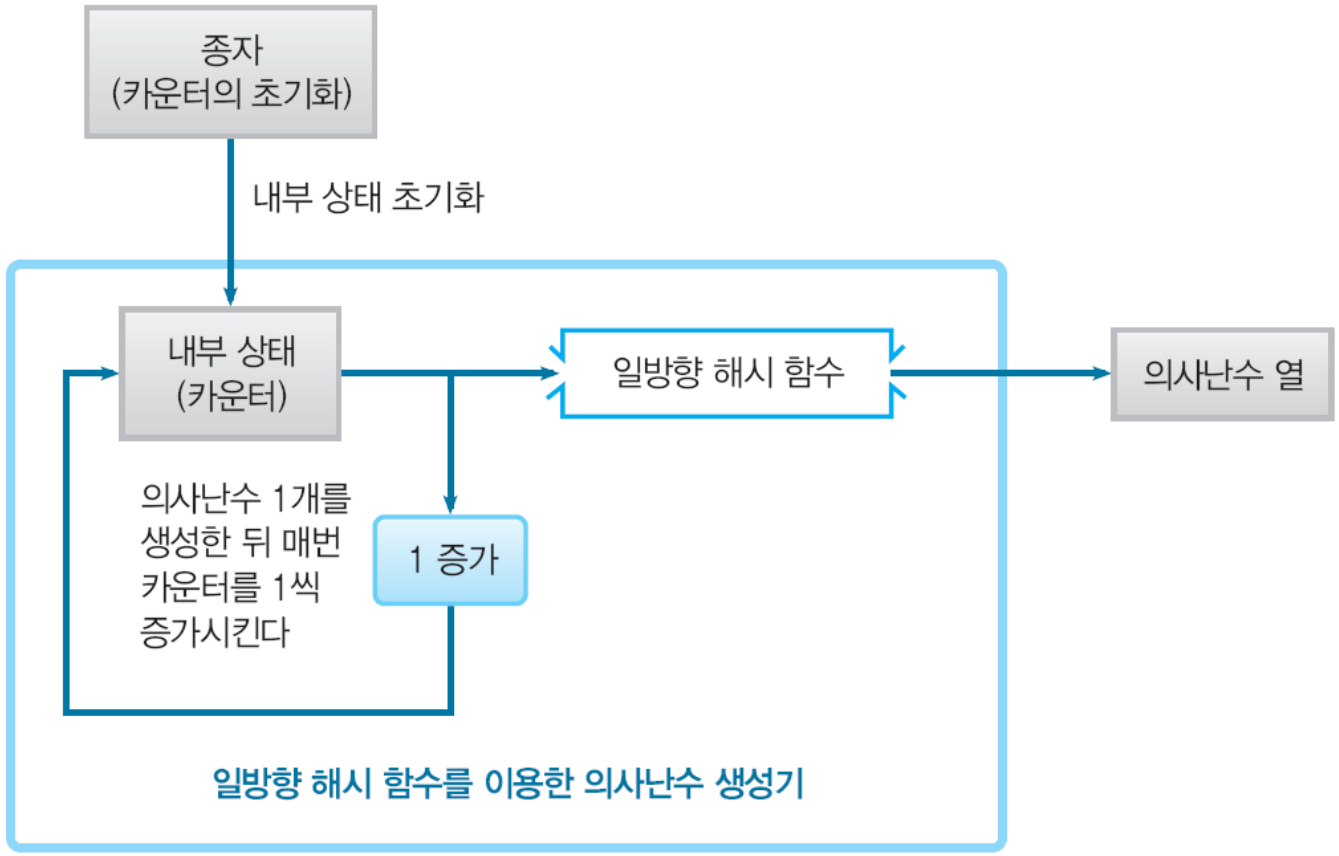


그림 13-5 • 일방향 해시 함수를 사용해서 의사난수 생성기를 만든다

잘못 만들어진 의사난수 생성기

- 다음과 같이 생성했다고 해보자
  1) 의사난수의 종자를 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

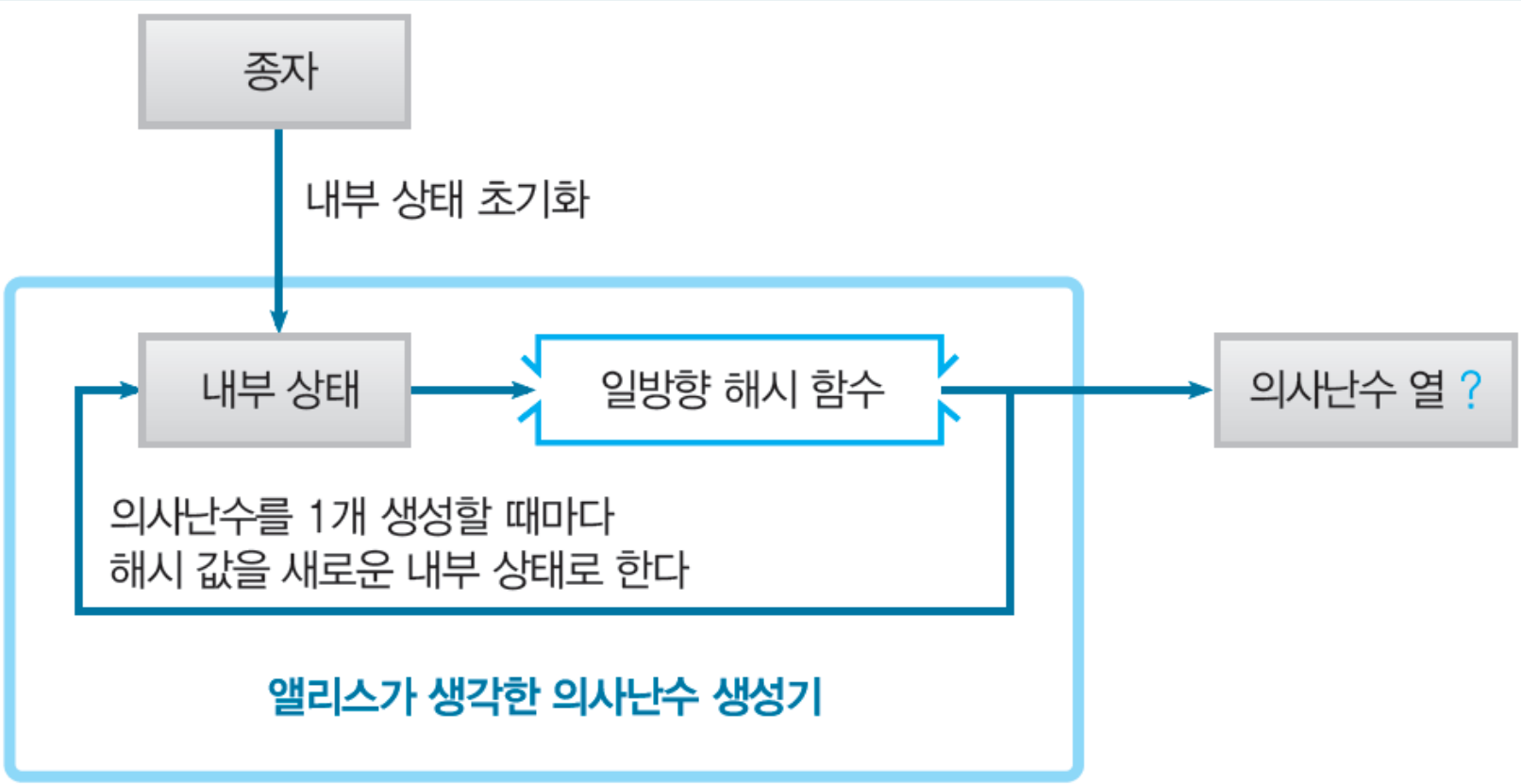


그림 13-6 • 앨리스가 생각한 의사난수 생성기의 약점을 찾아보자

왜 예측불가능성이 없나?

- 마지막에 출력한 의사난수의 해시 값을 취해서 다음 의사난수를 생성하므로 예측 불가능성을 갖지 않는다.
- 예측 불가능성을 갖기 위해서는 일방향 해시 함수의 일방향성을 사용하는 것이 포인트

칼럼 일방향 해시함수를 사용한 의사난수 생성기의 프로그램 개요

카운터 초기값이 의사난수 종자에 해당된다. counter값이 내부 상태에 해당된다.

```
counter = 카운터 초기값;
while(true) {
        의사난수 = 일방향해시함수로 counter 해시값을 얻는다;
        의사난수를 출력한다;
        counter 1을 증가시킨다;
}
```

4.4 암호를 사용하는 방법

- 암호를 사용해서「강한 의사난수」를 생성하는 의사난수 생성기를 만들 수 있다
- AES와 같은 대칭 암호나 RSA와 같은 공개키 암호 중 어느 것을 사용해도 무방
- 암호의 기밀성이 의사난수 생성기의 예측 불가능성을 보장

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

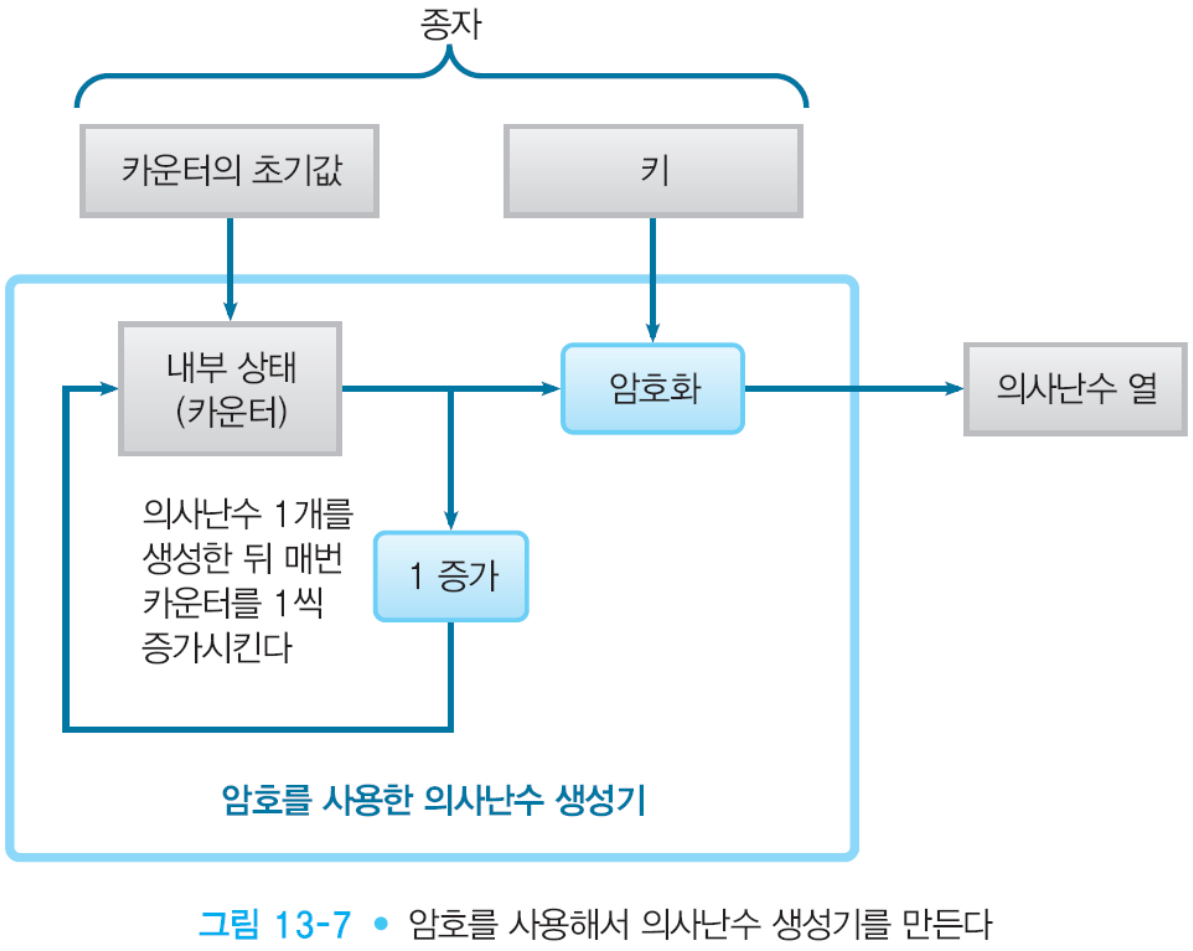


그림 13-7 • 암호를 사용해서 의사난수 생성기를 만든다

칼럼 암호를 사용한 의사난수생성기의 프로그램 개요

key값과 카운터 초기 값 쌍이 의사난수의『종자』에 해당된다. 카운터 값이 내부 상태에 해당된다.

key = 암호 키;

counter = 카운터 초기값;

```
while(true) {
        의사난수 = key를 사용해서 counter 를 암호화 한다;
        의사난수를 출력한다;
        counter를 1 증가시킨다;
}
```

4.5 ANSI X9.17

- 암호 소프트웨어 PGP에서 사용하는 의사 난수 생성기
  - ANSI X9.17
  - ANSI X9.31

ANSI X9.17 의사난수 생성 절차

1) 내부 상태를 초기화
2) 현재 시각을 암호화
3) 내부 상태와 암호화된 현재시각을 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 세팅
9) 절차(2)~(8)을 필요한 만큼의 의사난수가 얻어질 때까지 반복

ANSI X9.17 방법 의사난수 생성기

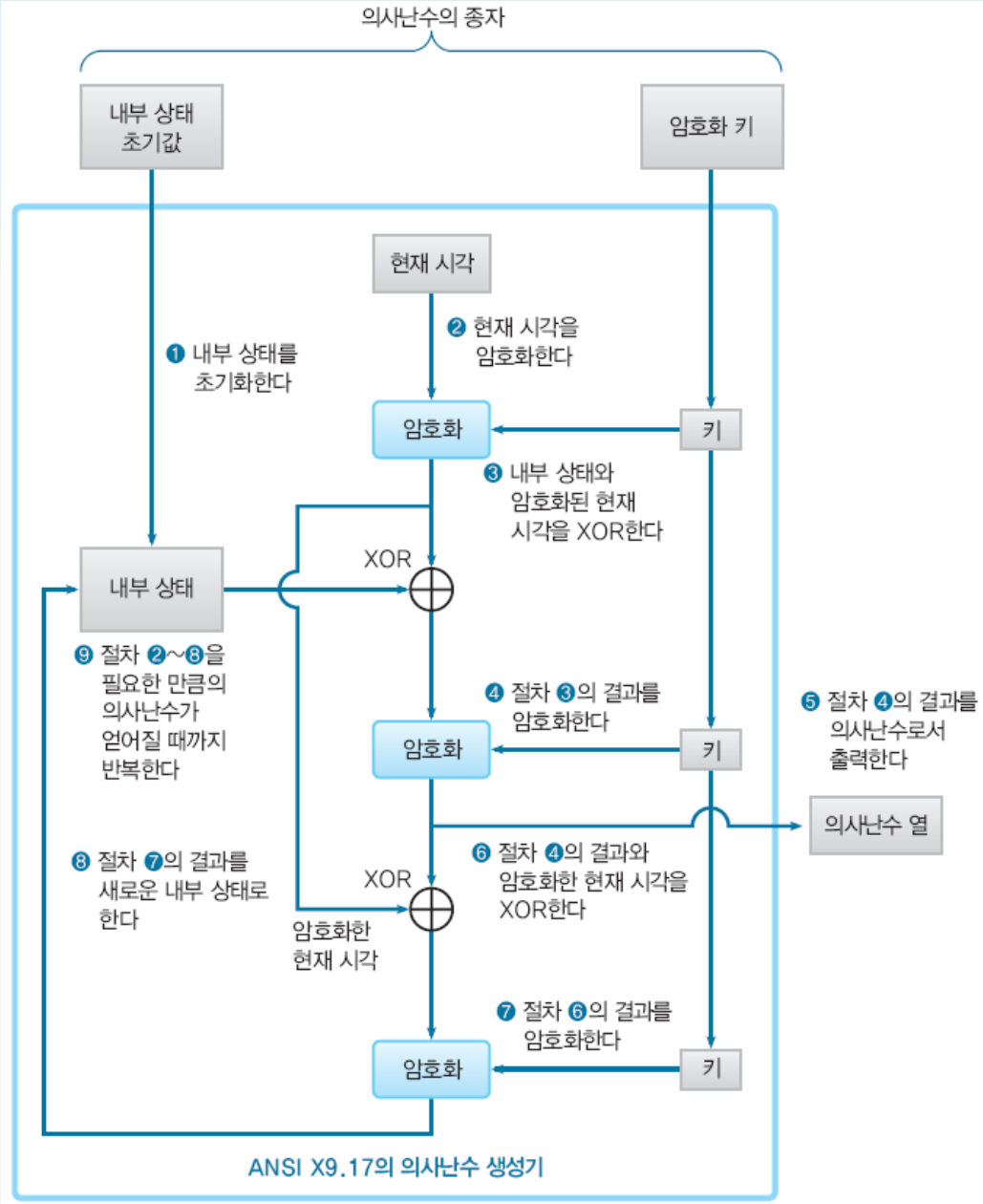


그림 13-8 • ANSI X9.17의 방법으로 의사난수 생성기를 만든다

왜 내부상태 추측을 못하나?

- 공격자는 의사난수로부터 역산해서 「내부상태와 암호화된 현재시각의 XOR」을 간파할 수 없다
  - 이유: 간파하기 위해서는 암호를 해독해야만 한다
- 공격자는 지금까지 출력된 의사난수열로부터 의사난수 생성기의 내부 상태를 추측 못 한다

칼럼 암호를 사용한 의사난수 생성기의 프로그램 개요

key 값과 내부 상태 초기값 쌍이 의사난수의 『종자』에 해당된다.
key=암호화 키;
내부상태=내부 상태의 초기화;
while(true) {

    암호화된 현재시각 = key를 사용해서 현재 시각을 암호화한다;
    의사난수 = key를 사용해서 『내부상태⊕암호화된 현재시각』을 암호화한다;
    의사난수를 출력한다;
    내부상태 = key를 사용해서 『의사난수⊕암호화된 현재시각』을 암호화한다;
}

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)
      - 유명한 의사 난수 생성 알고리즘이지만, 보안용으로는 사용하면 안 됨
    - 선형 합동법
      - 충분한 길이의 난수열을 관찰하면 앞으로 생성될 난수열이 예측 가능하므로 사용하면 안 됨

안전한 알고리즘

- JAVA
  - Java.util.Random이 있지만, 이것은 보안용으로는 사용하면 안 됨
  - **보안용**: java.security.SecureRandom클래스가 준비되어 있음
- Ruby
  - Random 클래스
  - **보안용**: SecureRandom 모듈 사용

Section 05
의사난수 생성기에 대한 공격

5.1 종자에 대한 공격

5.2 랜덤 풀에 대한 공격

5.1 종자에 대한 공격

- 종자의 중요성:
  - 의사난수의「종자」는 암호의「키」에 필적
  - 종자가 공격자에게 노출 되면, 그 의사난수 생성기가 생성한 모든 의사난수열은 공격자에게 노출
- 종자 선택:
  - 재현 불가능성을 갖는「진정한 난수」선택

5.2 랜덤 풀에 대한 공격

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

Quiz 3 난수의 기초지식

다음 문장 중 바른 것에는 O, 틀린것에는 X를 표시하시오.

(1) 의사 난수의 종자는 공격자에게 비밀로 해 둘 필요가 있다.

(2) 선형 합동법은 암호용의 의사난수 생성기로서 이용할 수 있다.

(3) 무작위성을 가지고 있는 의사난수 생성기라도 반드시 예측 불가능성을 갖는다고만 할 수 없다.