네트워크 보안 에센셜

- 2장 대칭암호와 메시지 기밀성 (2) -

임연주(yeonjoo@pel.smuc.ac.kr)

상명대학교 프로토콜공학연구실

목 차

스트림 암호 구조

- 스트림 암호 구조
  - 스트림 암호를 설계 하는 것
  - 블록 암호와 스트림 암호
    - 암호화 할 때 평문이 처리되는 단계에서 사용
      - 블록 암호: 블록 단위로 암호화 처리
      - 스트림 암호: 비트나 바이트 단위로 암호화 처리

스트림 암호 구조

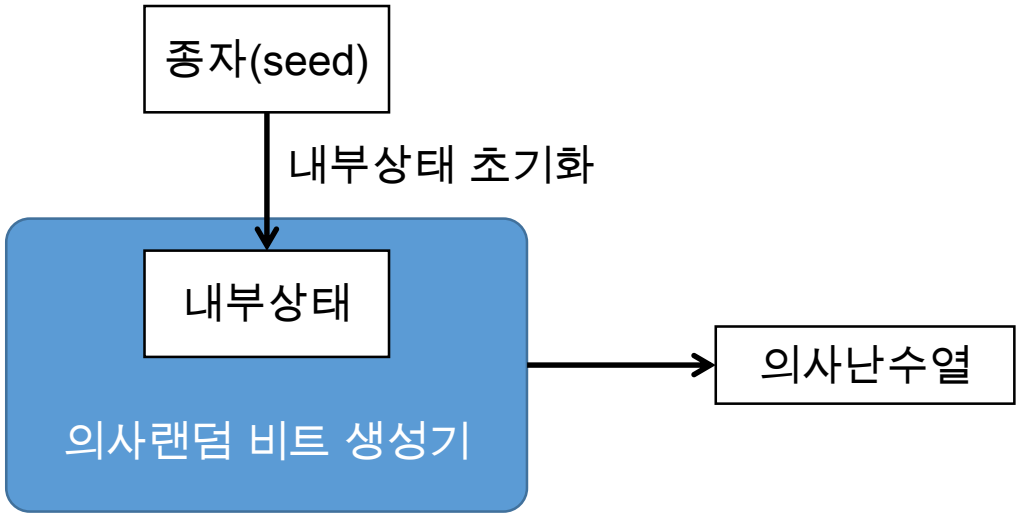
- 스트림 암호
  - 의사랜덤 비트 생성기를 이용
  - 평문의 바이트(비트)와 생성기에서 출력되는 바이트(비트)를 XOR하여 암호문의 바이트(비트) 얻음
  - 암호화하는 사용자와 복호화하는 사용자는 같은 키스트림을 가짐
    - 키스트림이란?
      - 의사랜덤 비트 생성기 (PRNG: pseudorandom number generator)의 출력 값
      - 평문과 XOR할 때 쓰임

스트림 암호 구조

- 의사랜덤 비트 생성기란?
  - 키스트림 생성기(keystream generator)라고도 함
  - 입력 값인 종자(Seed)를 기초로 의사 난수 열을 생성하는 것
    - 종자: 생성기의 내부상태를 초기화하기 위해 이용되는 랜덤한 비트 열
  - 같은 입력 값을 넣으면 동일한 결과 값이 나오는 결정적 알고리즘으로 이용해 비트 열 생성

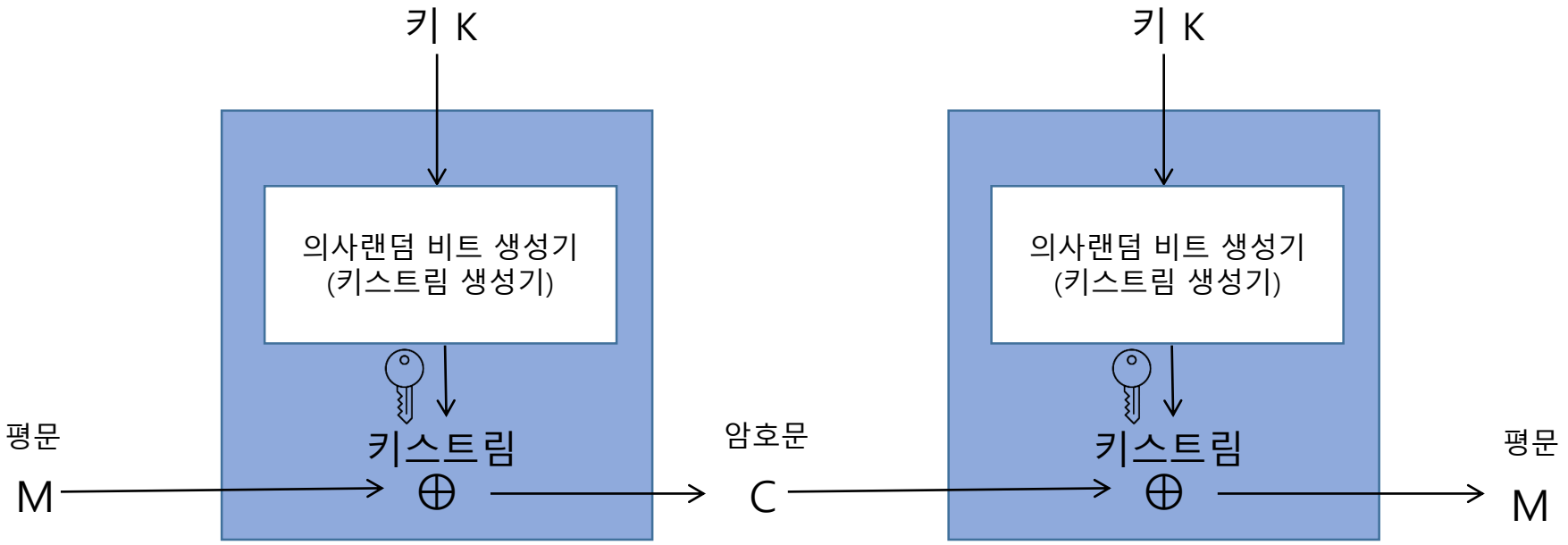
스트림 암호 구조

- 의사랜덤 비트 생성기의 구조



스트림 암호 구조

- 의사랜덤 비트 생성기(키스트림 생성기)를 이용한 스트림 암호 구조



스트림 암호 구조

- 스트림 암호 설계 시 고려사항

1. 암호열의 주기는 커야 함
   - 반복되는 스트림이 없을수록 해독하기 어려움

2. 키스트림을 랜덤 스트림화
   - 키스트림이 랜덤하게 구성 될수록 암호문은 더 랜덤 해지고 해독은 그만큼 더 어려워짐

3. 키의 길이가 길어야 함
   - 전수 공격을 이용한 해독을 방지
     - 전수 공격이란?
       - 암호 해독을 위한 공격유형 중 하나
       - 공격자가 해독 해야 할 암호문만 가지고 다양한 통계적인 테스트를 하면서 암호를 해독하는 것

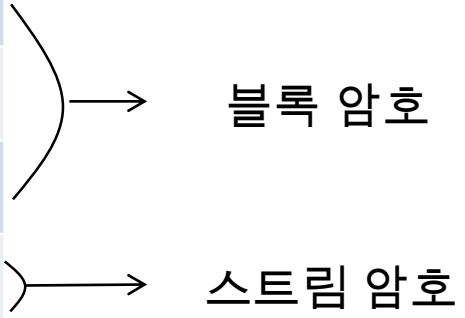
스트림 암호 구조

- 스트림 암호 구조의 장단점
  - 장점
    - 동일한 크기의 키를 사용하는 블록 암호만큼 안전함
    - 블록 암호에 비해 간단하고 빠름
  - 단점
    - 공격자가 키스트림이 노출되면, 그 키스트림을 이용해 생성한 모든 암호문은 공격자에게 노출됨

스트림 암호 구조

- 대칭 블록 암호와 실행속도 비교

암호 알고리즘	키 길이(비트)	속도(Mbps)
DES	56	9
3중 DES	168	3
RC2	다양한 길이	0.9
RC4	다양한 길이	45



목 차

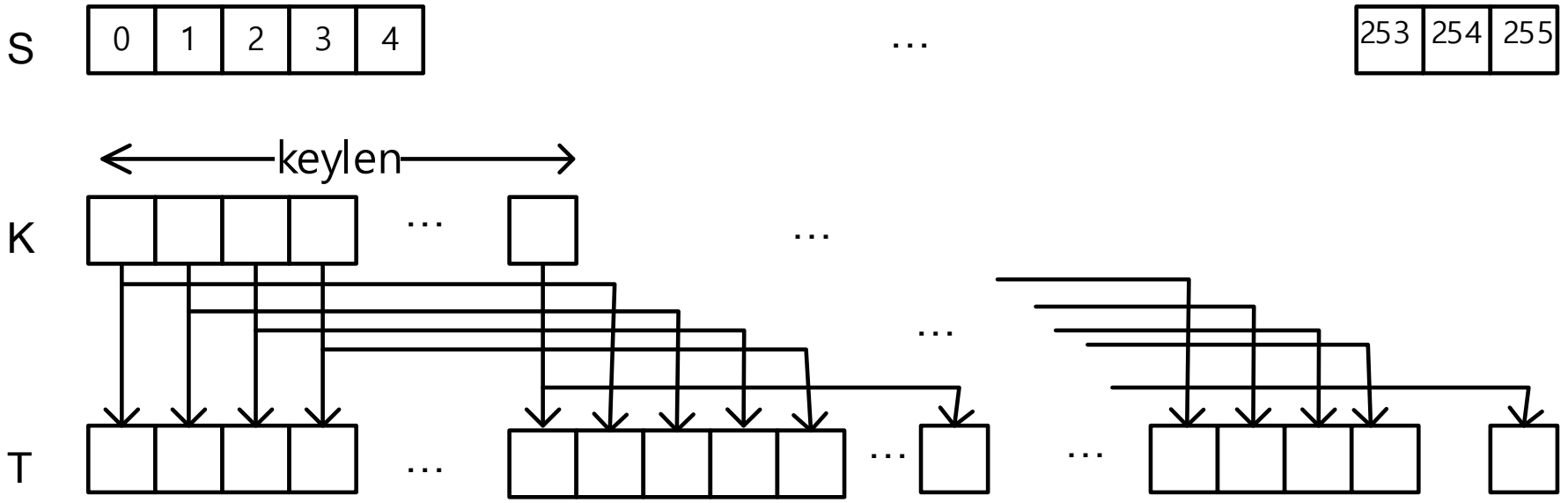
RC4 알고리즘

- RC4 알고리즘의 특징
  - 바이트 단위를 이용하여 다양한 키 사이즈를 갖는 스트림 암호 방식
  - 랜덤 치환 방식을 기반으로 함

- RC4 알고리즘 구현
  1. 백터 S의 초기화
  2. 백터 S의 초기 치환
  3. 스트림 생성

RC4 알고리즘

- S의 초기화



- 우선 백터 S의 성분은 0부터 255까지 오름차순으로 저장시킴
- 임시 백터 T를 만든 뒤, K의 keylen 인덱스 까지의 값을 T가 꽉 찰 때 까지 반복해서 복사

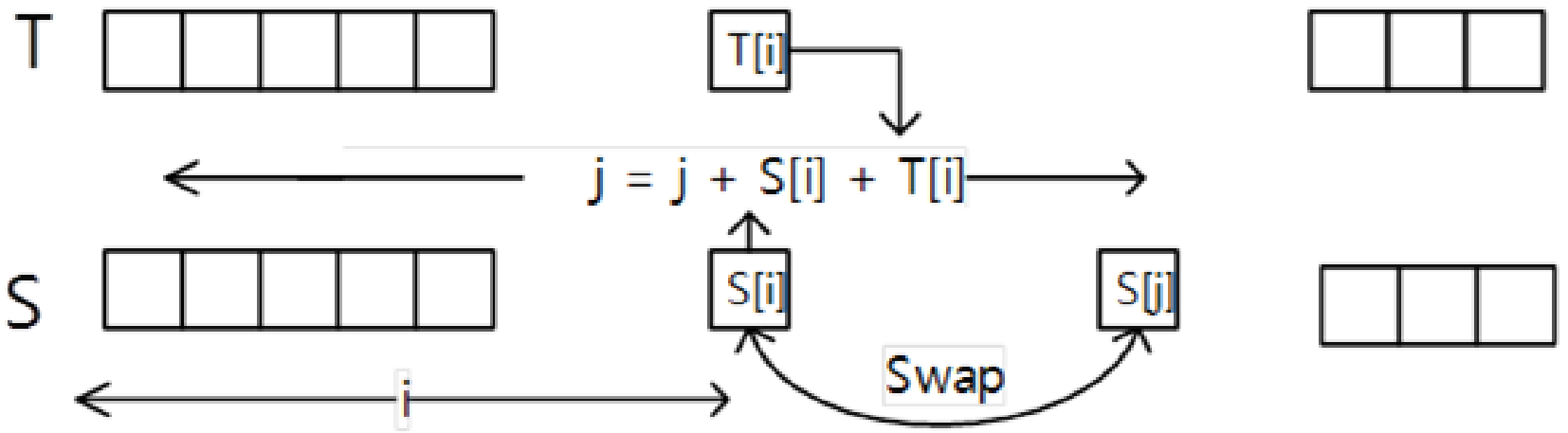
RC4 알고리즘

- S의 초기화
  - ❖알고리즘으로 구현

```
for i = 0 to 255 do
    S[i] = i;                          // 0부터 255까지 오름차순으로 저장
    T[i] = K[i mod keylen];  // 나머지 연산으로 반복해서 복사 구현
```

RC4 알고리즘

- S의 초기 치환



- 백터 T의 저장된 값을 이용해 S를 다른 바이트와 교환

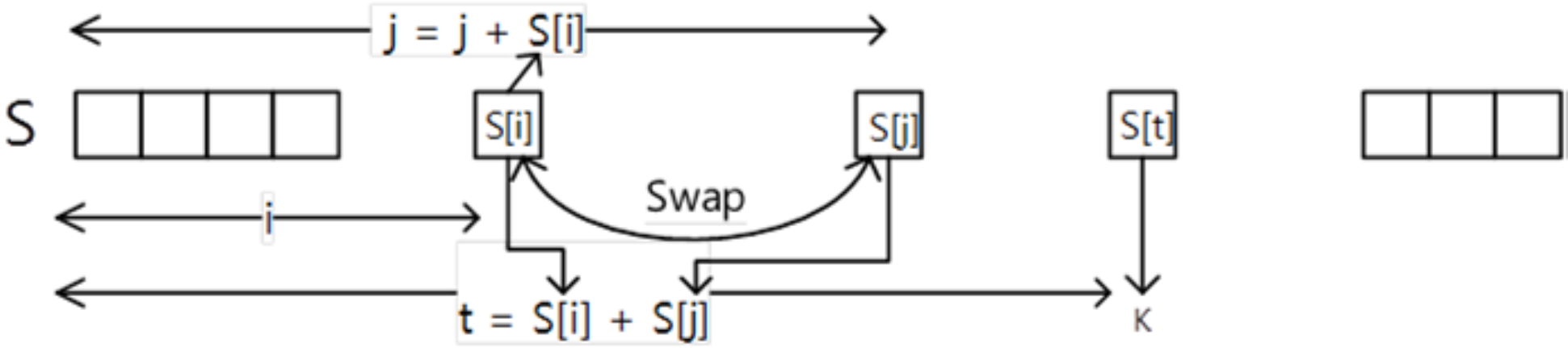
RC4 알고리즘

- S의 초기 치환
  - ❖알고리즘으로 구현

```
        j = 0;
for i = 0 to 255 do
        j = (j + S[i] + T[i] ) mod 256; // T를 이용해 바꿀 인덱스를 계산
        Swap (S[i], S[j]);     // S[i]와 S[j] 위치 교환
```

RC4 알고리즘

- 스트림 생성



- 암호화: 변수 K와 평문의 다음 바이트를 XOR한 뒤 리턴
- 복호화: 변수 k와 암호문의 다음 바이트를 XOR한 뒤 리턴

RC4 알고리즘

- 스트림 생성

❖알고리즘으로 구현

```
    i ,j = 0;
while (true)
        i = (i + 1) mod 256; // i를 1만큼 증가
        j = (j + S[i]) mod 256;
        Swap (S[i], S[j]);    // s[i]와 s[j] 위치 교환
        t = (S[i] + S[j]) mod 256;
        k = S[t];

return text ^ k;
```

목 차

- 스트림 암호와 RC4
  - 스트림 암호 구조
  - RC4 알고리즘
- 암호 블록 운용모드
  - 전자 코드북 모드
  - 암호 블록 체인 모드
  - 암호 피드백 모드
  - 출력 피드백 모드
  - 카운터 모드

암호 블록 운용 모드

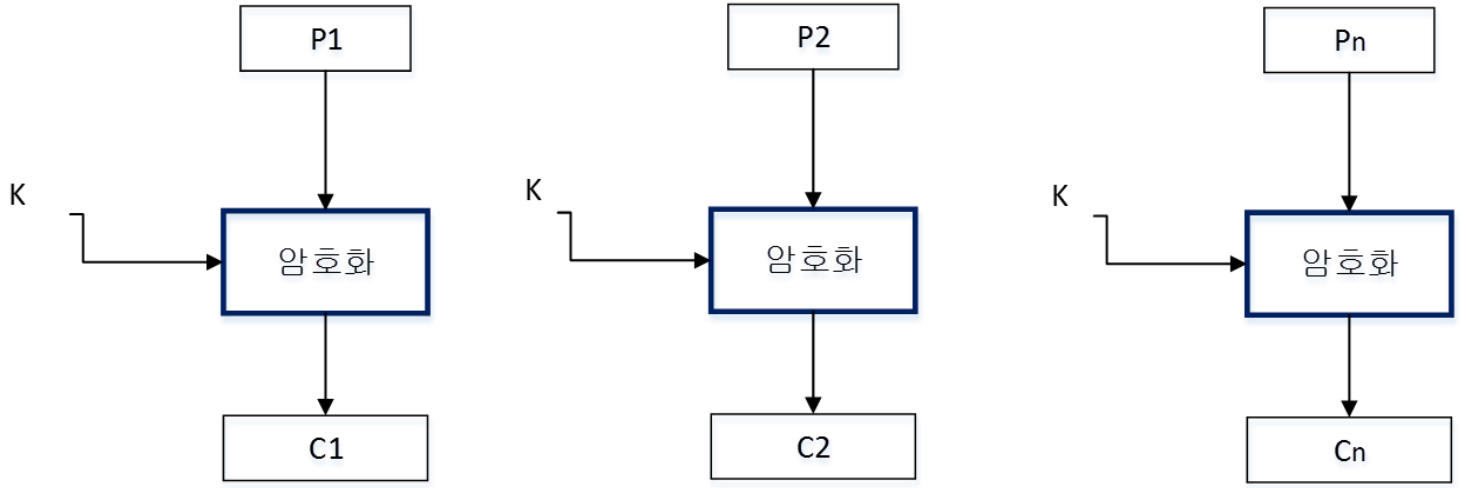
- 암호 블록 운용 모드는 왜 쓰일까?
  - 평문의 길이가 블록 암호의 블록 크기보다 큰 경우에 적용할 수 있는 방법
  - DES나 AES와 같은 블록 암호를 사용하여 다양한 크기의 데이터를 암호화하는 방식

목 차

- 스트림 암호와 RC4
  - 스트림 암호 구조
  - RC4 알고리즘
- 암호 블록 운용모드
  - 전자 코드북 모드
  - 암호 블록 체인 모드
  - 암호 피드백 모드
  - 보충) 출력 피드백 모드
  - 카운터 모드

전자 코드북 모드

- 전자 코드북 모드(electronic codebook (ECB) mode)



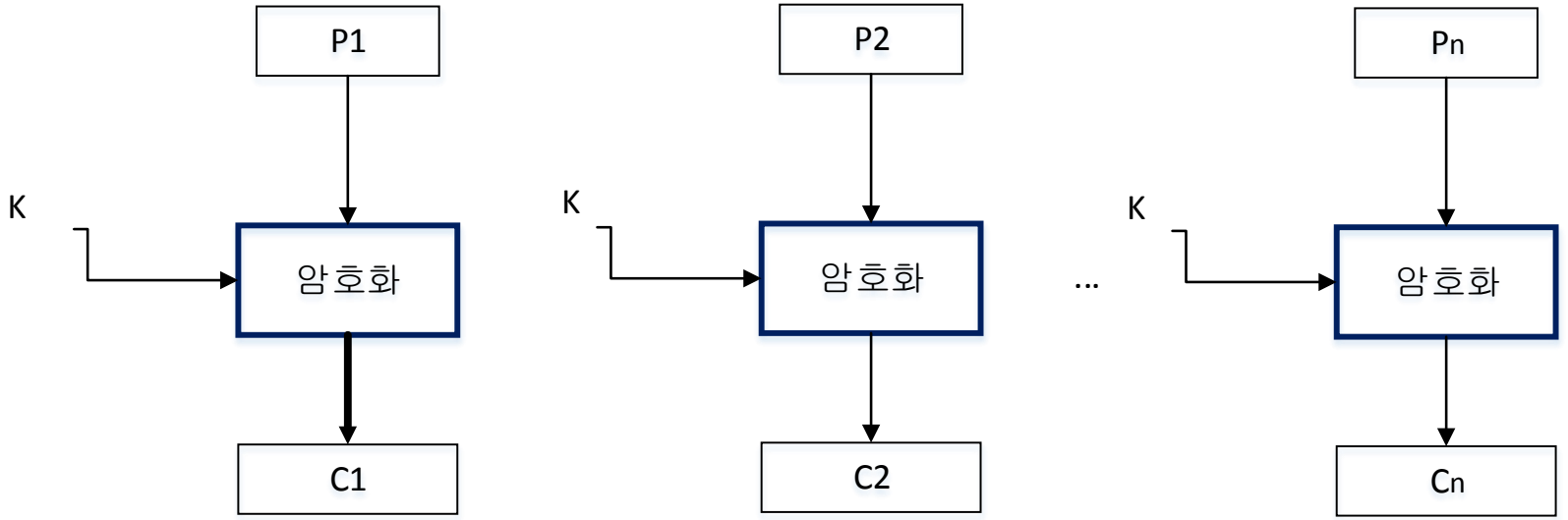
- b비트 이상의 메시지를 b비트 블록으로 나누고 암호화하는 모드
- 마지막 비트가 b비트 미만이면 나머지 비트를 채운 후 진행
  → 패딩
  - ✓패딩 : 부족한 길이만큼 '0'으로 채우거나 임의의 비트들로 채워 넣는 것

전자 코드북 모드

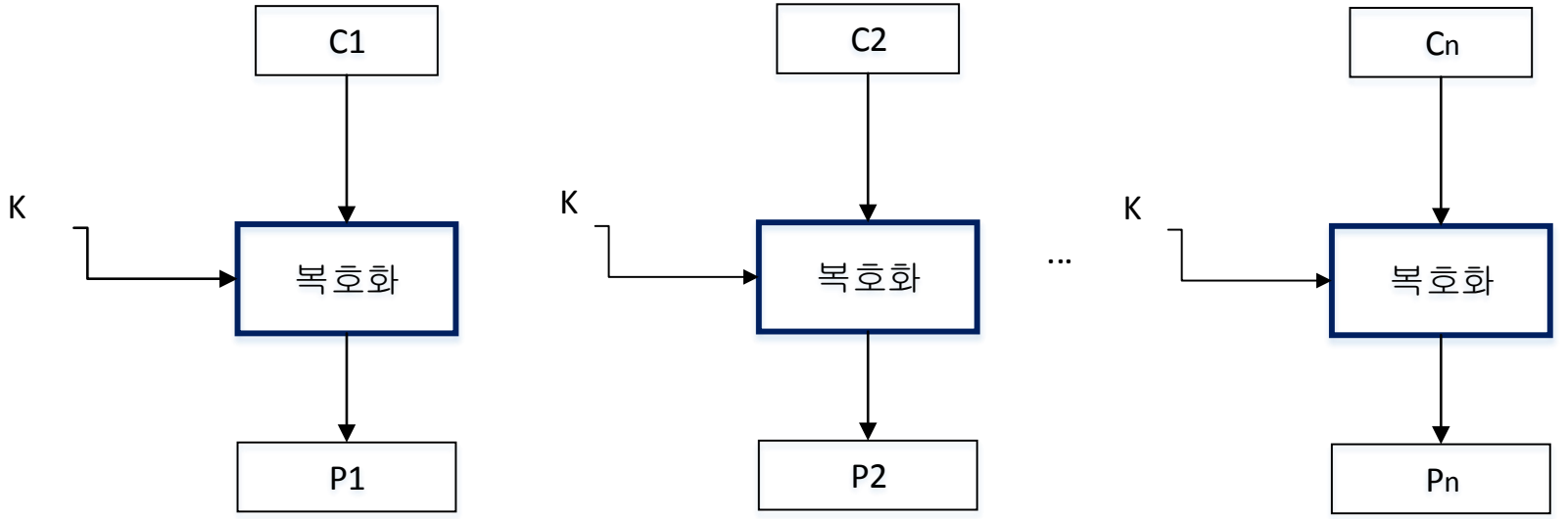
- ECB의 장단점
  - 장점
    - 간단한 구조, 빠른 처리
    - 오류 확산 없음
    - 개별적으로 암호화, 복호화 진행 → 병렬 처리 가능
  - 단점
    - 같은 내용의 평문 블록은 같은 암호문이 출력
    → 암호 해독이 쉬워짐 암호문
    - 블록의 삭제나 교체에 의한 평문의 조작이 가능
    - 패딩 필요

전자 코드북 모드

(ㄱ) 암호화



(ㄴ) 복호화

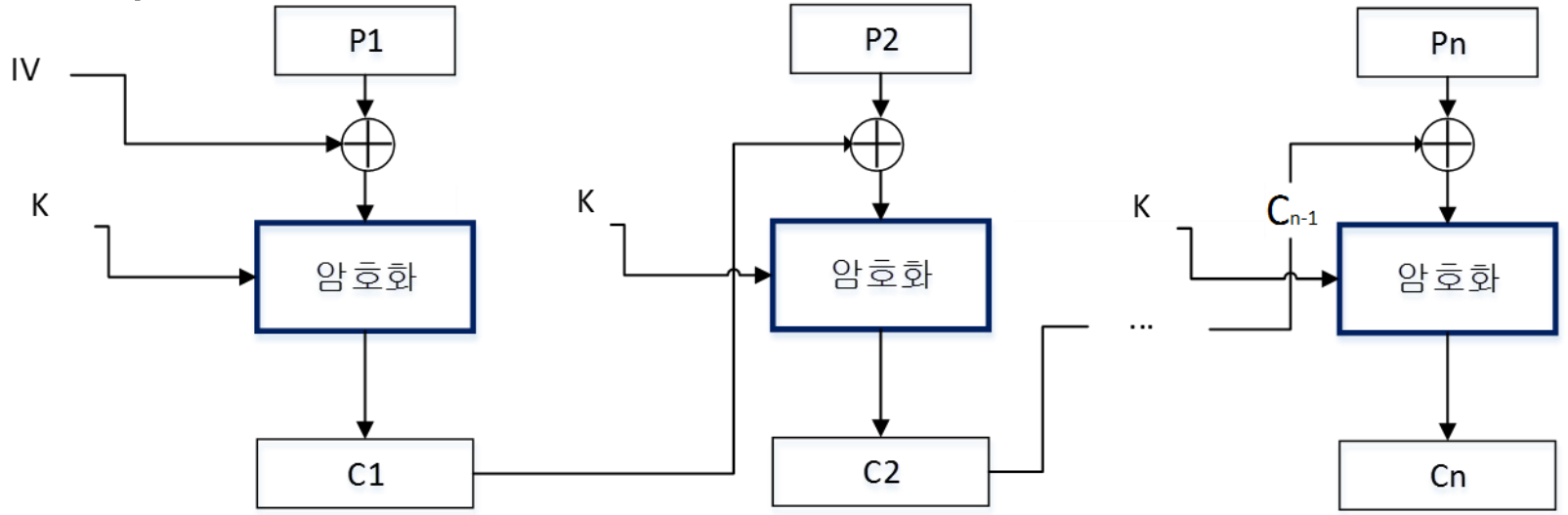


목 차

- 스트림 암호와 RC4
  - 스트림 암호 구조
  - RC4 알고리즘
- 암호 블록 운용모드
  - 전자 코드북 모드
  - 암호 블록 체인 모드
  - 암호 피드백 모드
  - 출력 피드백 모드
  - 카운터 모드

암호 블록 체인 모드

- 암호 블록 체인 모드 (cipher block chaining(CBC) mode)



  - 이전 암호문 블록의 출력 값과 현재 평문을 XOR한 결과 값이 암호화 과정에서 입력으로 들어감
  - 첫 번째 암호화에 적용할 초기화 백터(IV : Initialization Vector)값은 송수신자가 미리 인지

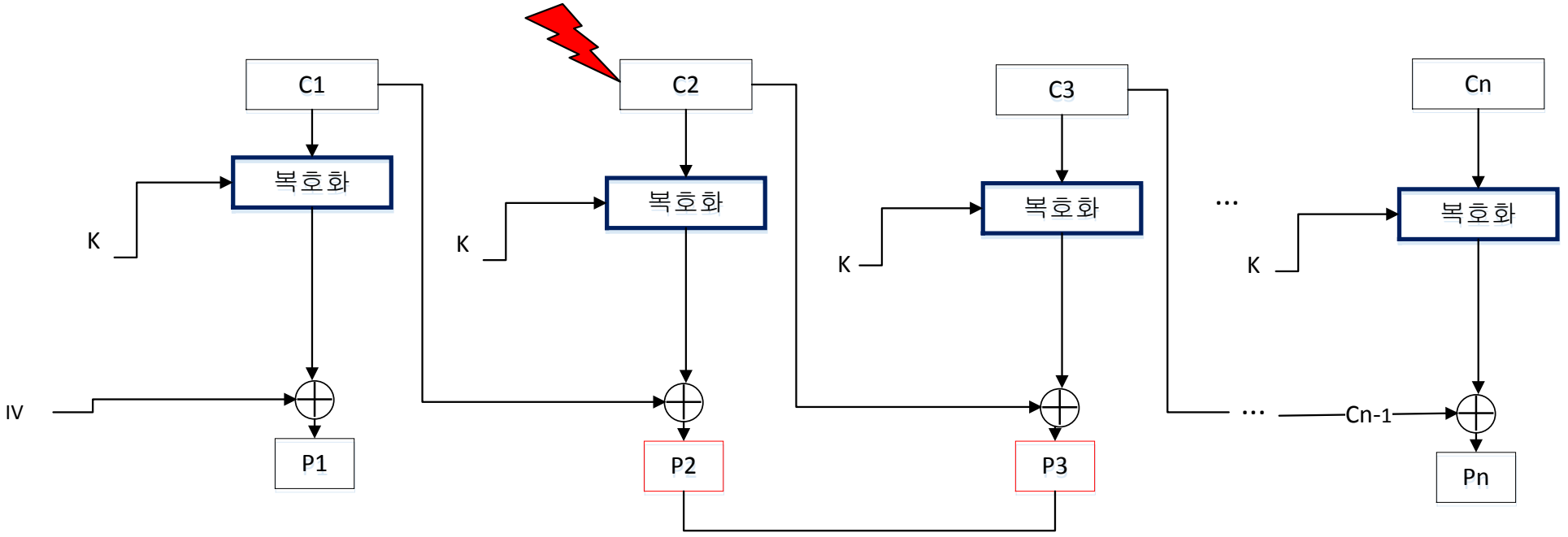
초기화 백터

✓초기화 백터 (IV : Initialization Vector) 란?

- 처음에는 1 단계 앞의 암호문 블록이 존재하지 않으므로 대신할 블록을 말함
- 키와 같이 보호해야 함
- 실제 환경에서 IV의 기밀성이 아니라 무결성이 중요 (만약 공격자가 전송되는 IV의 한 비트를 변경시킨다면 수신자는 제대로 된 평문을 얻을 수 없기 때문)

암호 블록 체인 모드

- 오류확산
  - 암호문 블록이 1개 파손되었을 때, 복호화 과정에서 손상된 암호문을 복호한 평문과 그 다음 평문에만 영향을 끼침

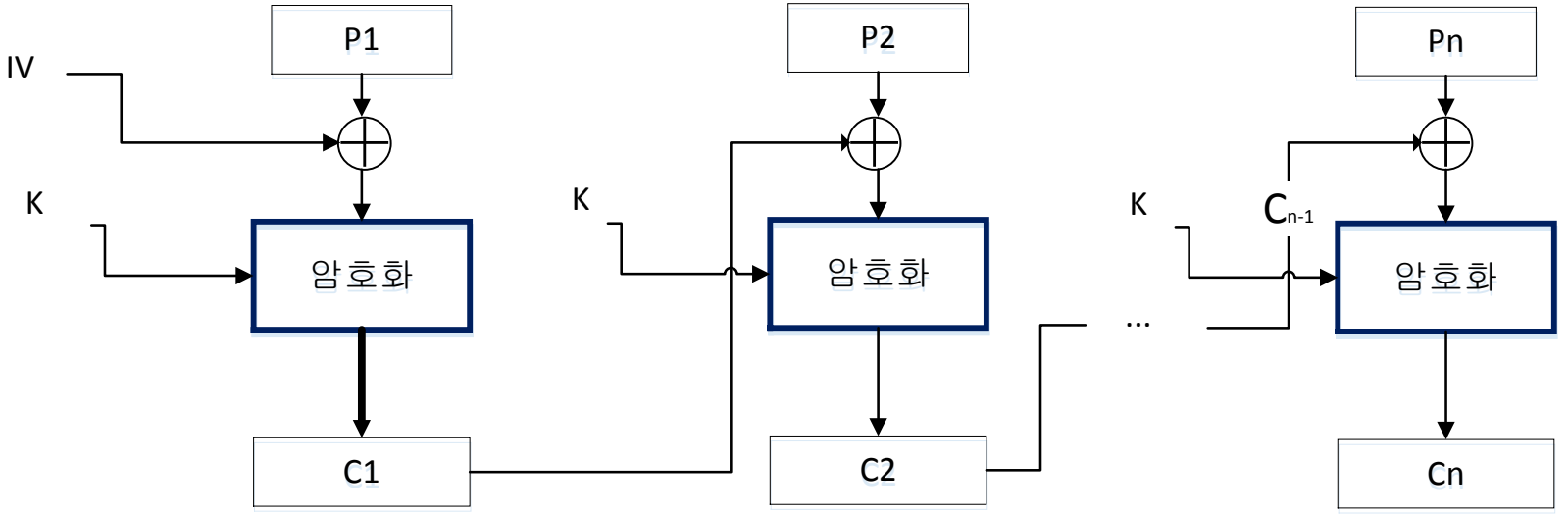


암호 블록 체인 모드

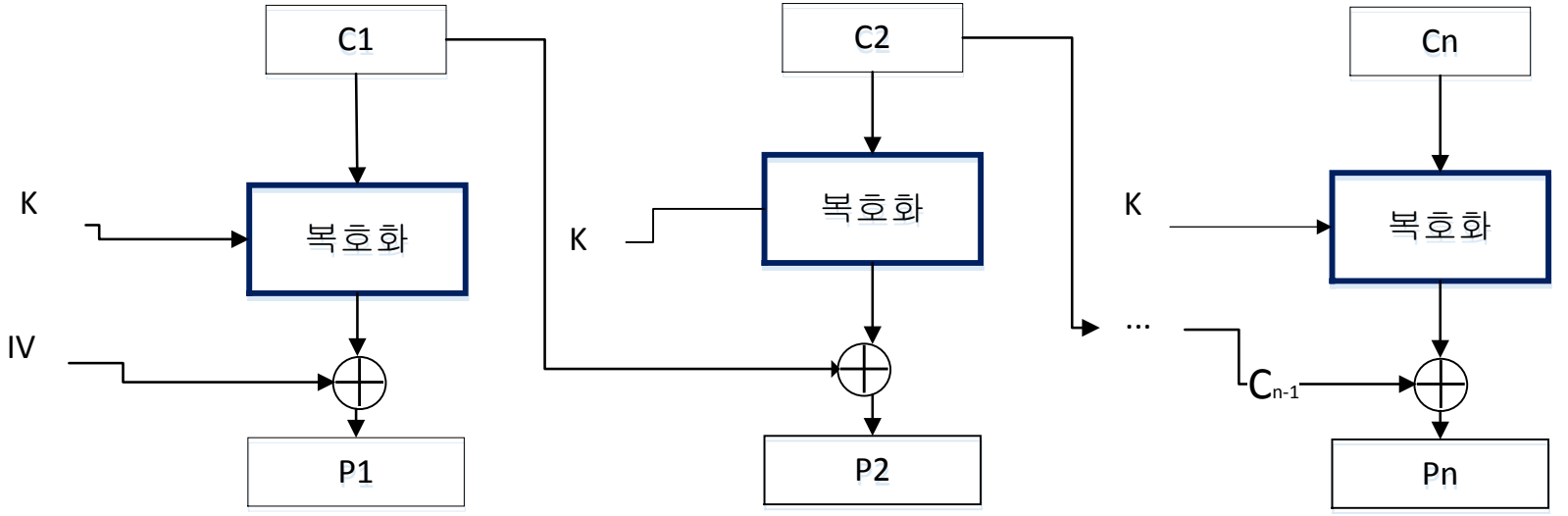
- CBC의 장단점
  - 장점
    - 같은 내용의 평문 블록을 넣어도 다른 암호문이 출력
    - ECB보다 복잡한 구조 → 해독하기 어려움
    - 평문 블록 패턴이 보이지 않음
  - 단점
    - 암호화 과정은 병렬처리가 되지 않음
    - 도중의 평문 블록만을 뽑아내서 암호화할 수는 없음
    - 패딩 필요
    - 오류 확산

암호 블록 체인 모드

(ㄱ) 암호화



(ㄴ) 복호화

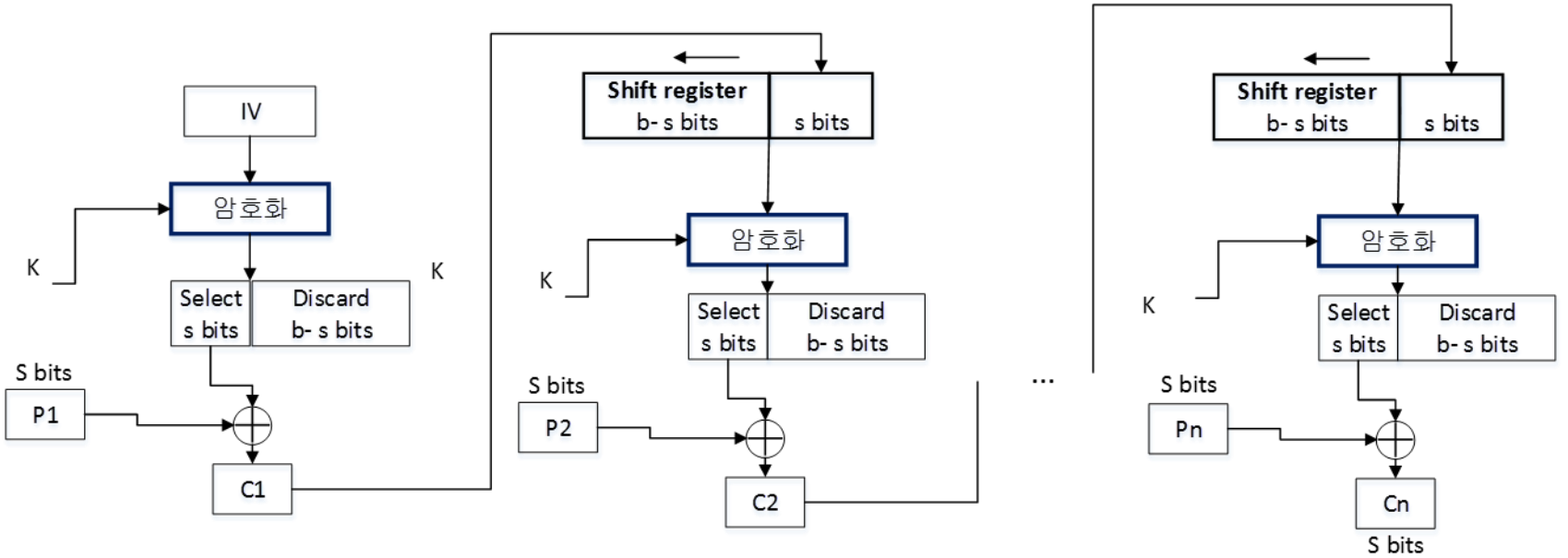


목 차

- 스트림 암호와 RC4
  - 스트림 암호 구조
  - RC4 알고리즘
- 암호 블록 운용모드
  - 전자 코드북 모드
  - 암호 블록 체인 모드
  - 암호 피드백 모드
  - 출력 피드백 모드
  - 카운터 모드

암호 피드백 모드

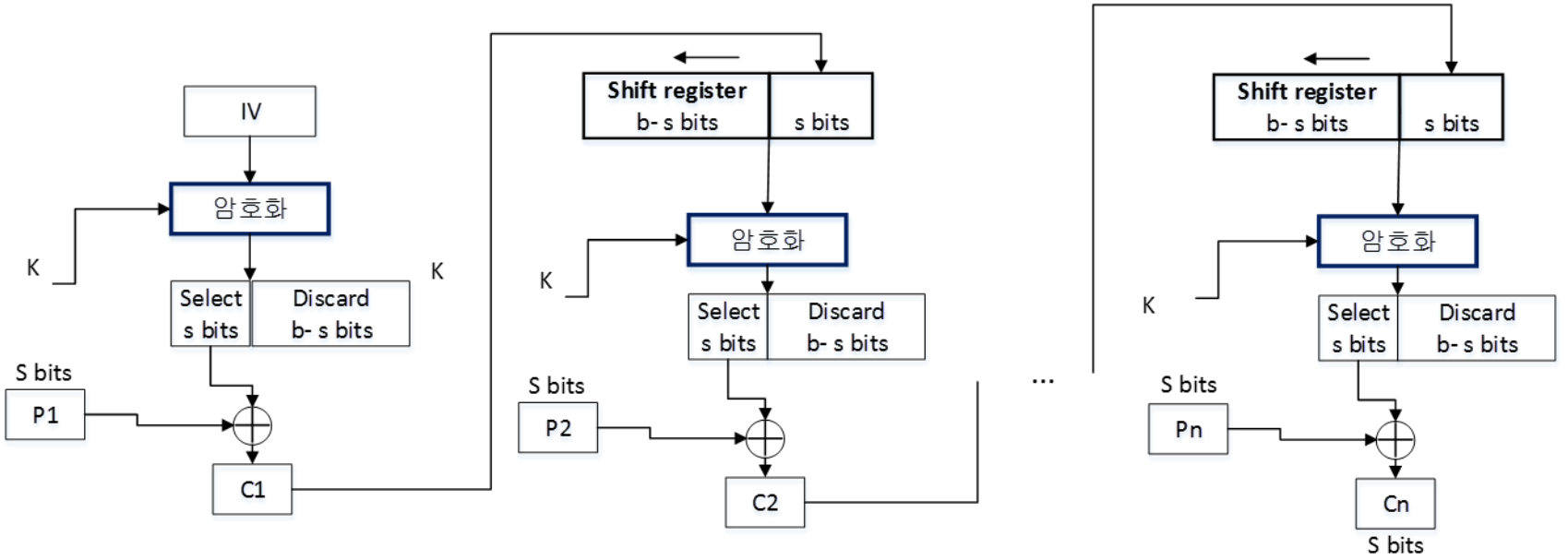
- 암호 피드백 모드(cipher feedback(CFB) mode)



- 암호화 함수 출력의 가장 왼쪽 s개 비트와 평문(s비트)을 XOR 연산
- 이전 단계 암호 블록을 암호화 과정의 입력 값으로 한 뒤, 오른쪽으로 시프트
- 모든 평문이 암호화 될 때 까지 반복

암호 피드백 모드

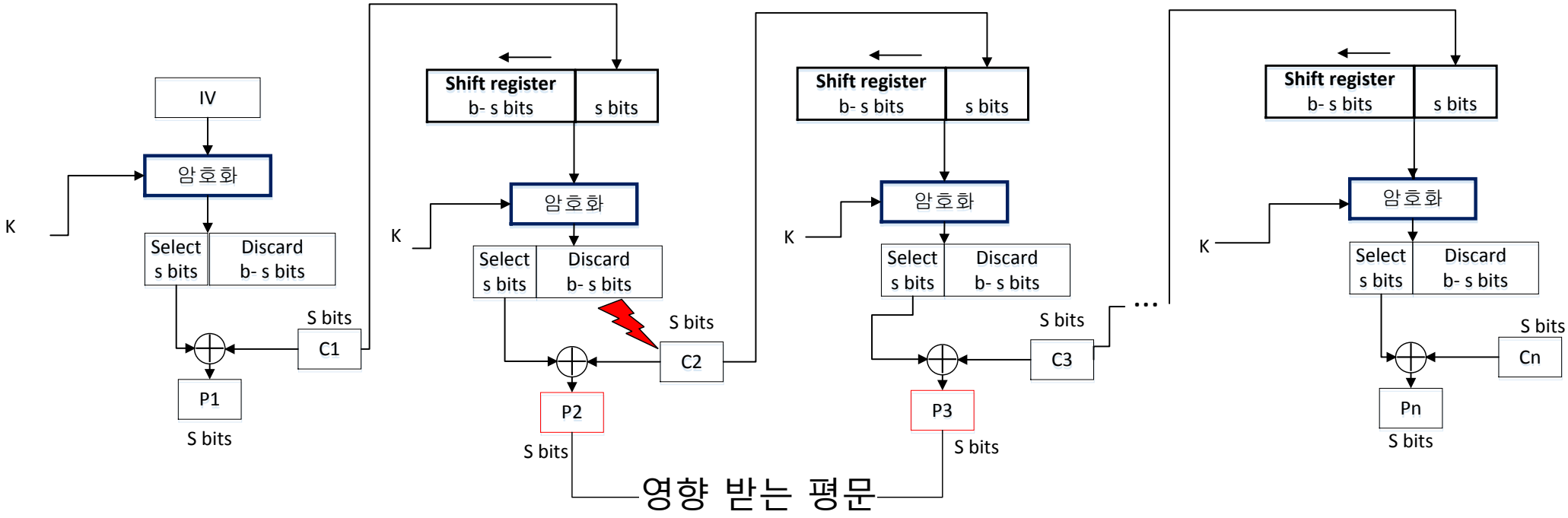
- 암호 피드백 모드(cipher feedback(CFB) mode)



- 암호 피드백 모드의 IV는 64비트 시프트 레지스터
- 블록 암호화를 스트림 암호화처럼 구성해 평문과 암호문의 길이가 같음

암호 피드백 모드

- 오류확산
  - 복호화 과정에서 C2이 손상 되었을 경우 손상된 암호문을 복호한 평문과 그 다음 평문에만 영향을 끼침

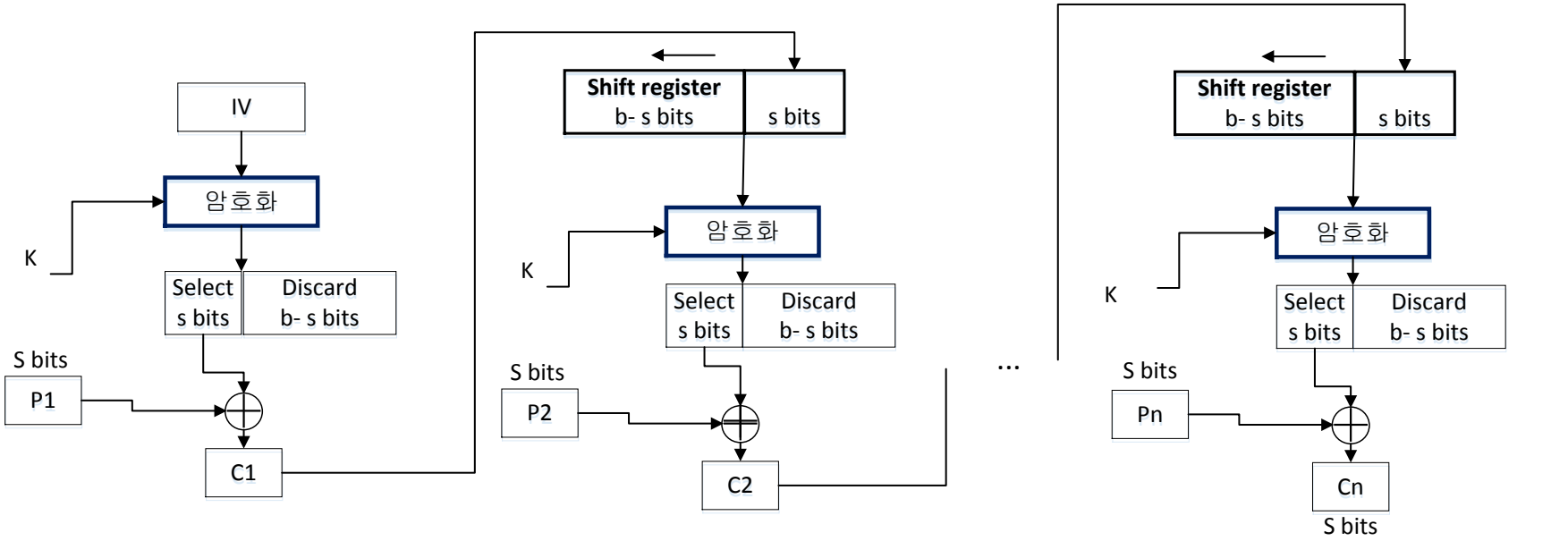


암호 피드백 모드

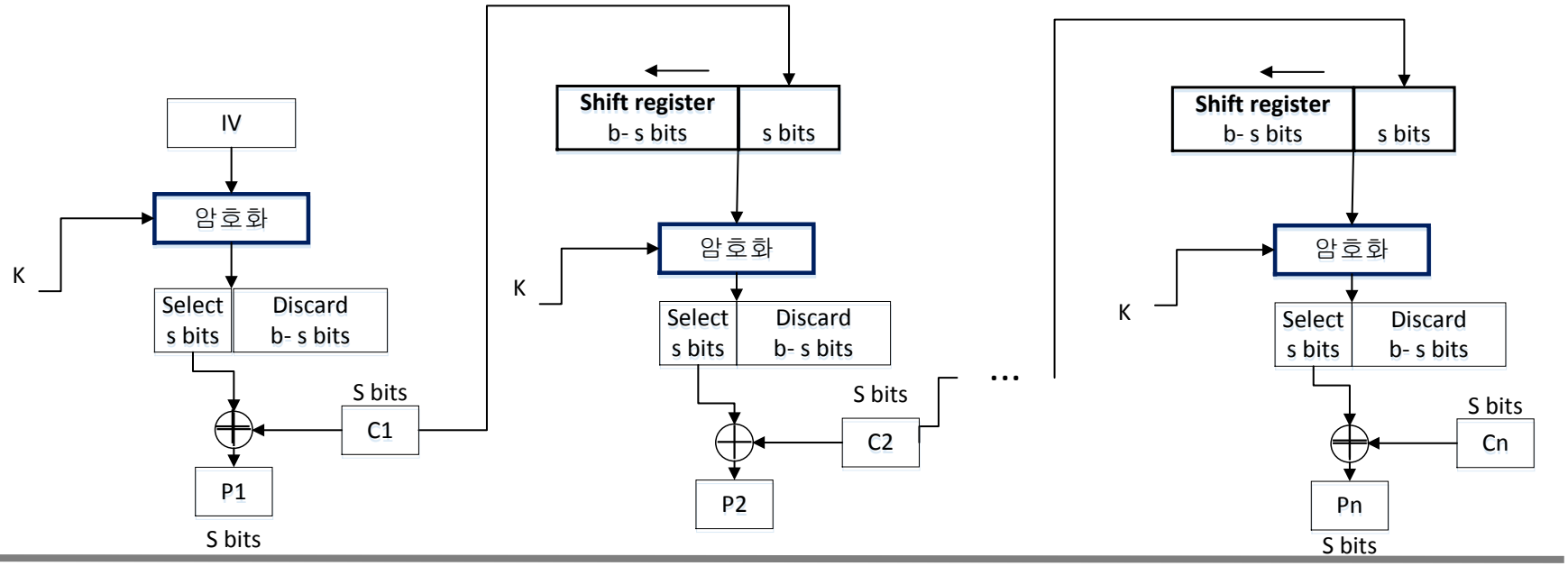
- CFB의 장단점
  - 장점
    - CBC와 달리 패딩 필요 없음
    - 복호화 과정에서 병렬처리 가능
    - 암호화 알고리즘과 복호화 알고리즘이 같음
  - 단점
    - 암호화에서 병렬처리 할 수 없음
    - 오류 확산
    - 도중의 평문 블록만을 뽑아내서 암호화할 수는 없음

암호 피드백 모드

(ㄱ) 암호화



(ㄴ) 복호화

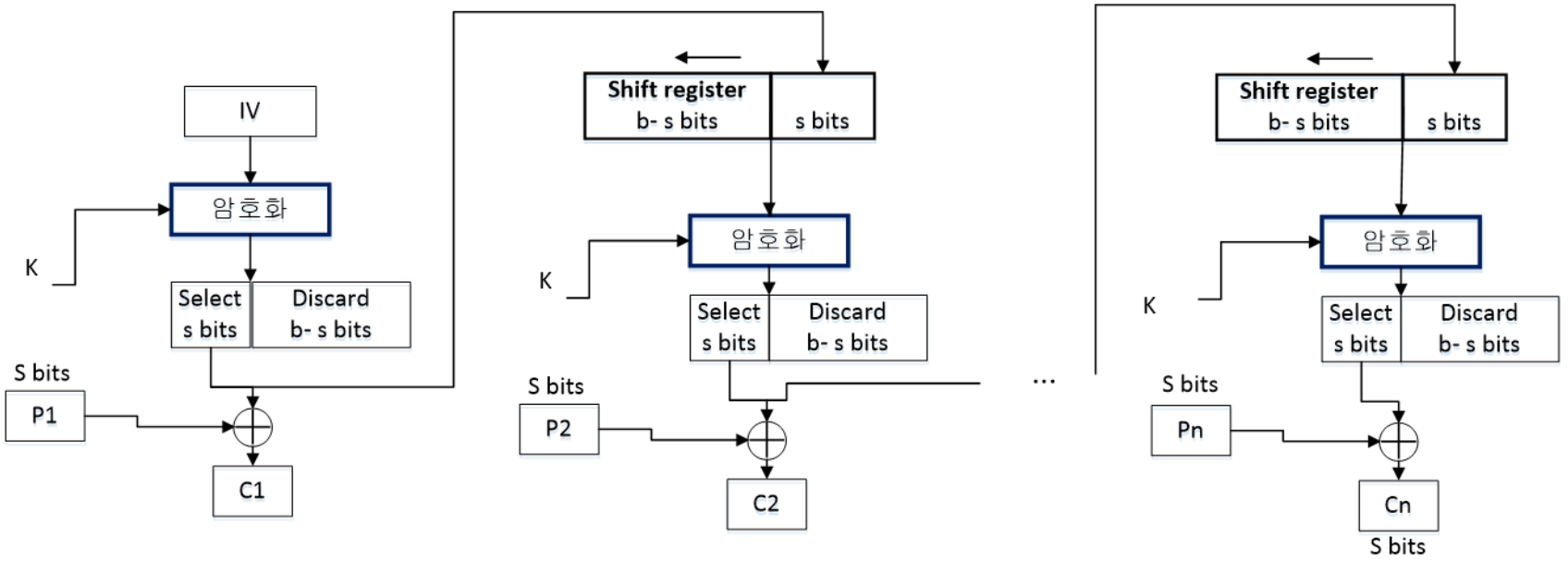


목 차

- 스트림 암호와 RC4
  - 스트림 암호 구조
  - RC4 알고리즘
- 암호 블록 운용모드
  - 전자 코드북 모드
  - 암호 블록 체인 모드
  - 암호 피드백 모드
  - 출력 피드백 모드
  - 카운터 모드

출력 피드백 모드

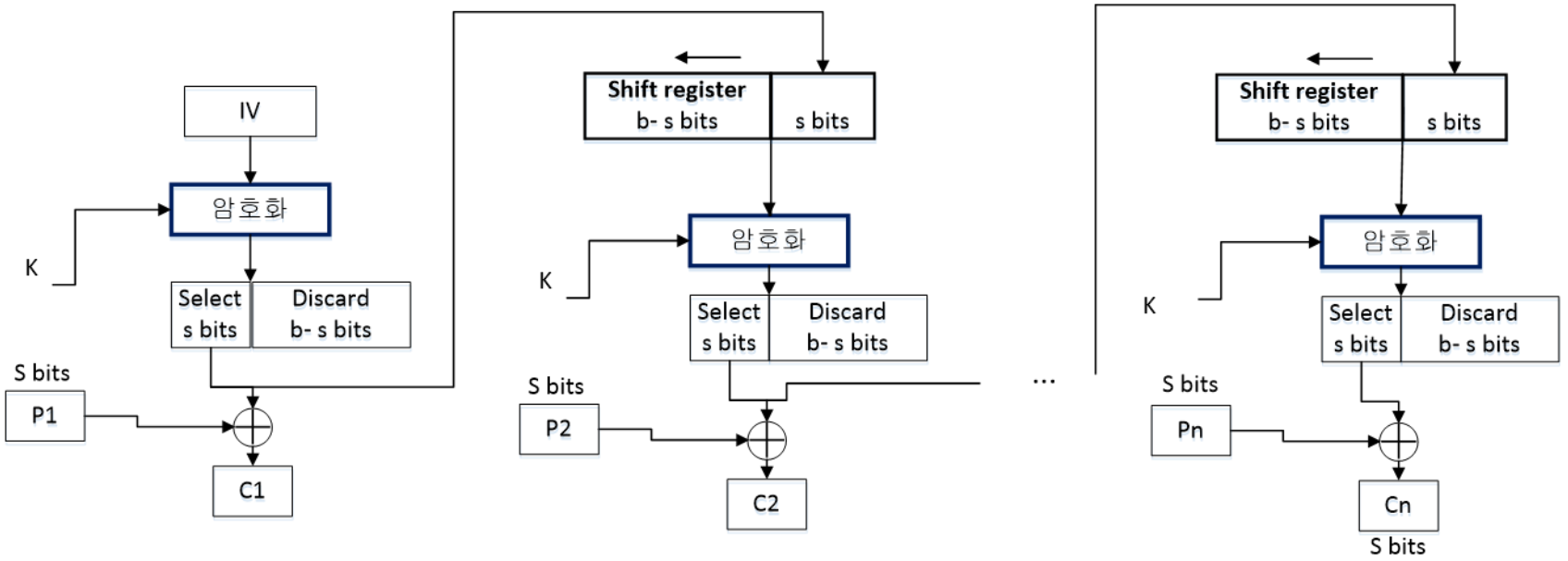
- 출력 피드백 모드 (output feedback(OFB) mode)



- 암호화 함수에서 출력을 다음 암호화 과정 입력으로 넣음
- 평문 블록과 암호 알고리즘의 출력을 XOR해서 암호문 블록을 만들어냄

출력 피드백 모드

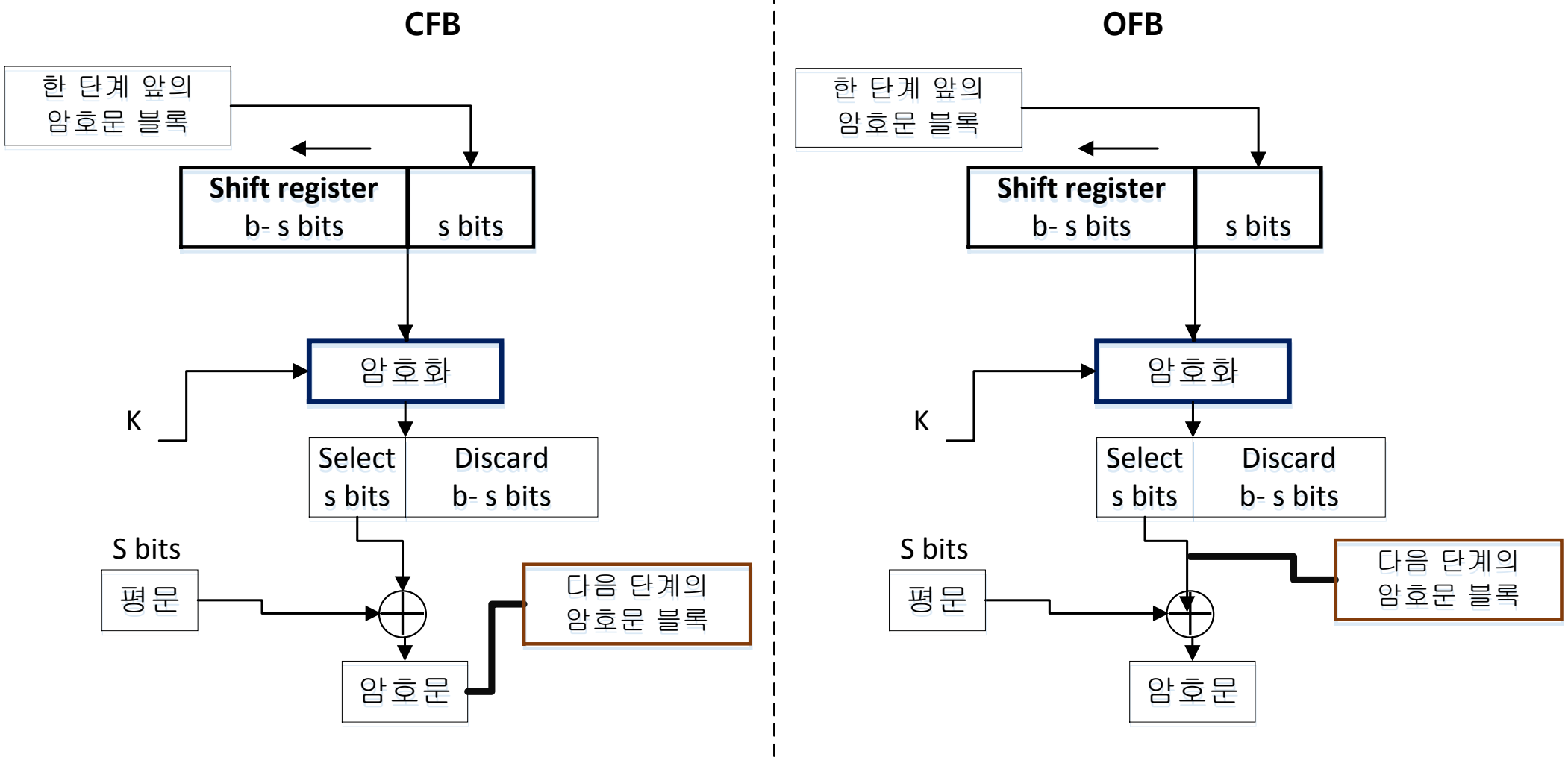
- 출력 피드백 모드 (output feedback(OFB) mode)



- 블록 암호화를 스트림 암호화처럼 구성해 평문과 암호문의 길이가 같음
- 평문 블록은 암호 알고리즘에 의해 직접 암호화되고 있는 것은 아님

출력 피드백 모드

- CFB와 OFB 비교

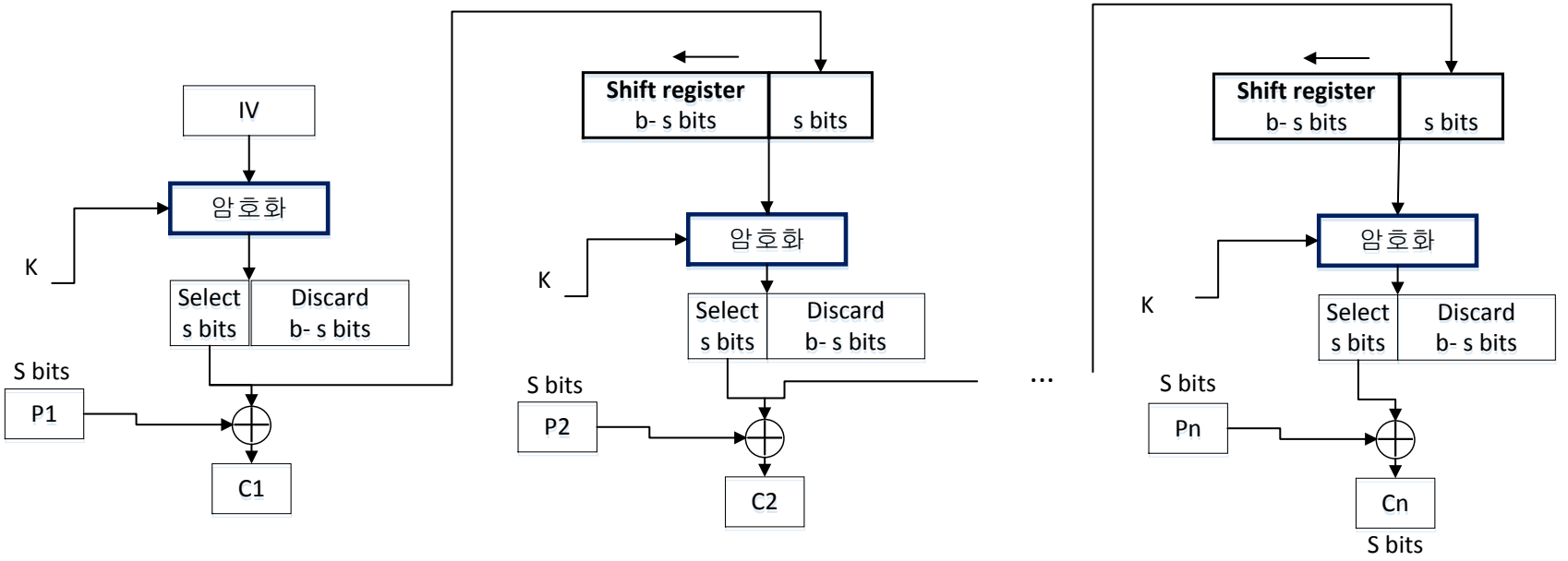


출력 피드백 모드

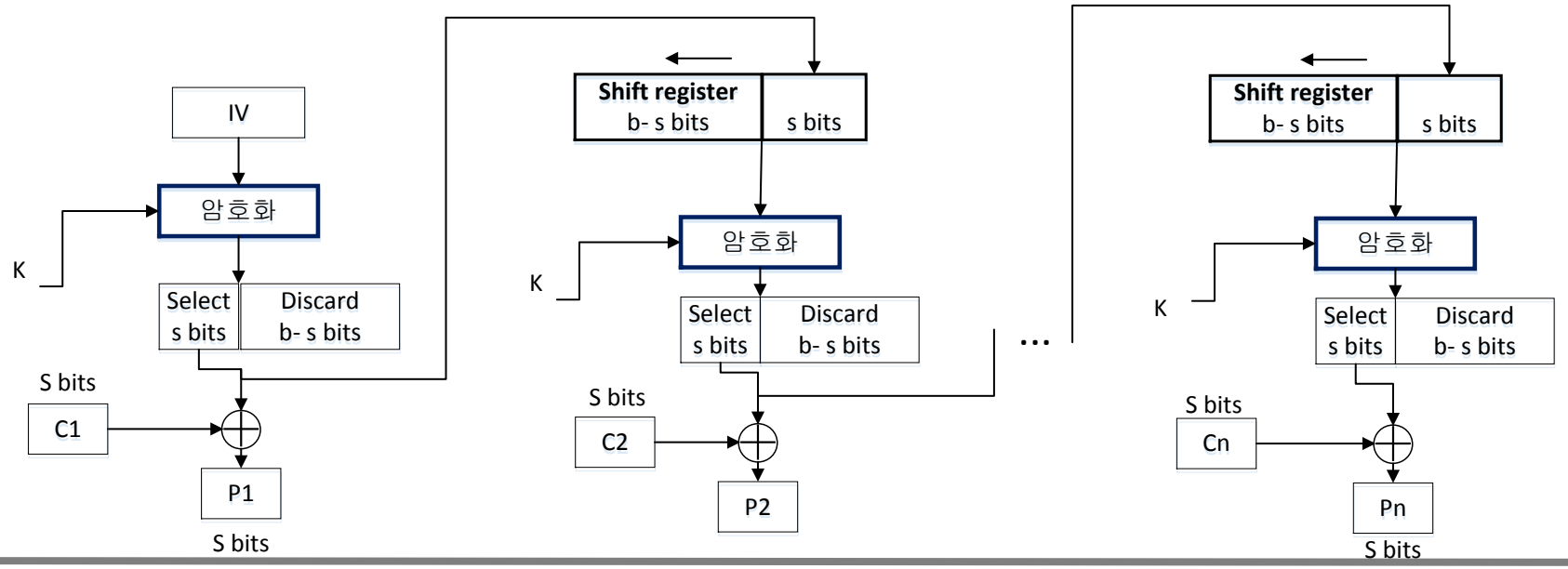
- OFB의 장단점
  - 장점
    - CFB와 달리 오류 확산 없음
    - CBC와 달리 패딩 필요 없음
    - 암호화와 복호화가 같은 구조
  - 단점
    - 암호화, 복호화 둘 다 병렬처리가 되지 않음
    - 도중의 평문 블록만을 뽑아내서 암호화할 수는 없음

출력 피드백 모드

(ㄱ) 암호화



(ㄴ) 복호화

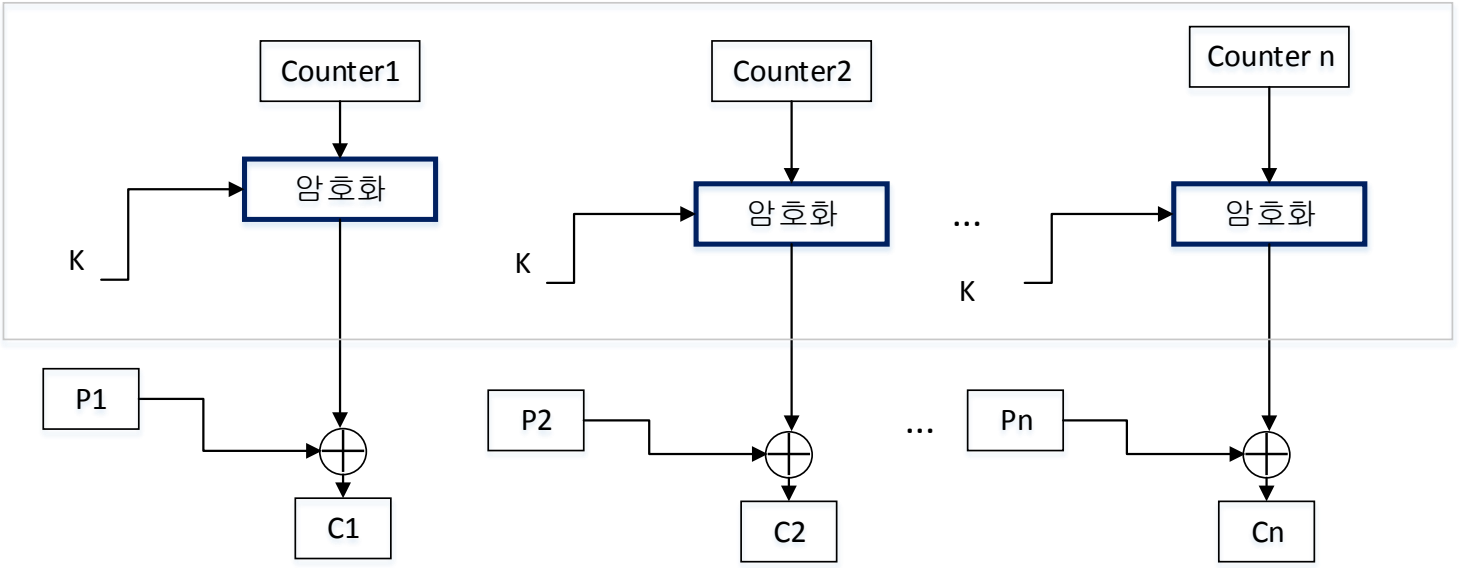


목 차

- 스트림 암호와 RC4
  - 스트림 암호 구조
  - RC4 알고리즘
- 암호 블록 운용모드
  - 전자 코드북 모드
  - 암호 블록 체인 모드
  - 암호 피드백 모드
  - 출력 피드백 모드
  - 카운터 모드

카운터 모드

- 카운터 모드 (counter(CTR) mode)



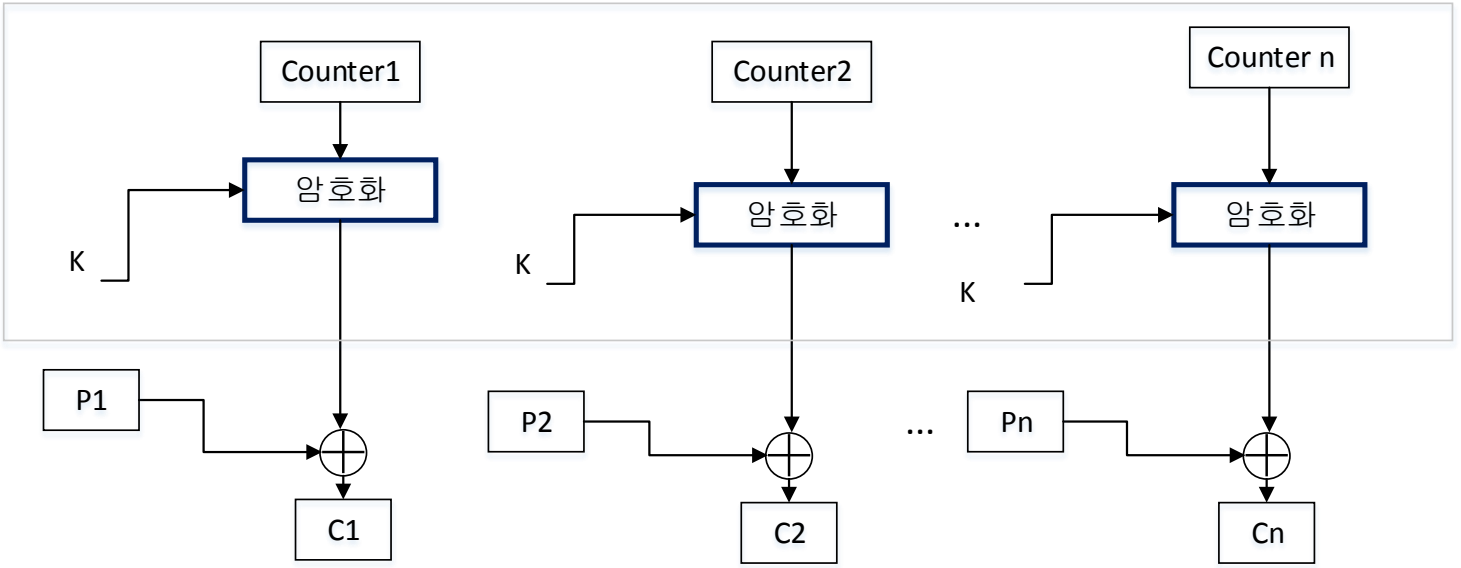
- 평문 블록과 동일한 크기의 카운터를 사용
- 카운터가 블록마다 다른 값이 들어감
- 카운터를 암호화하고 평문 블록과 XOR하여 암호문 생성

카운터 모드

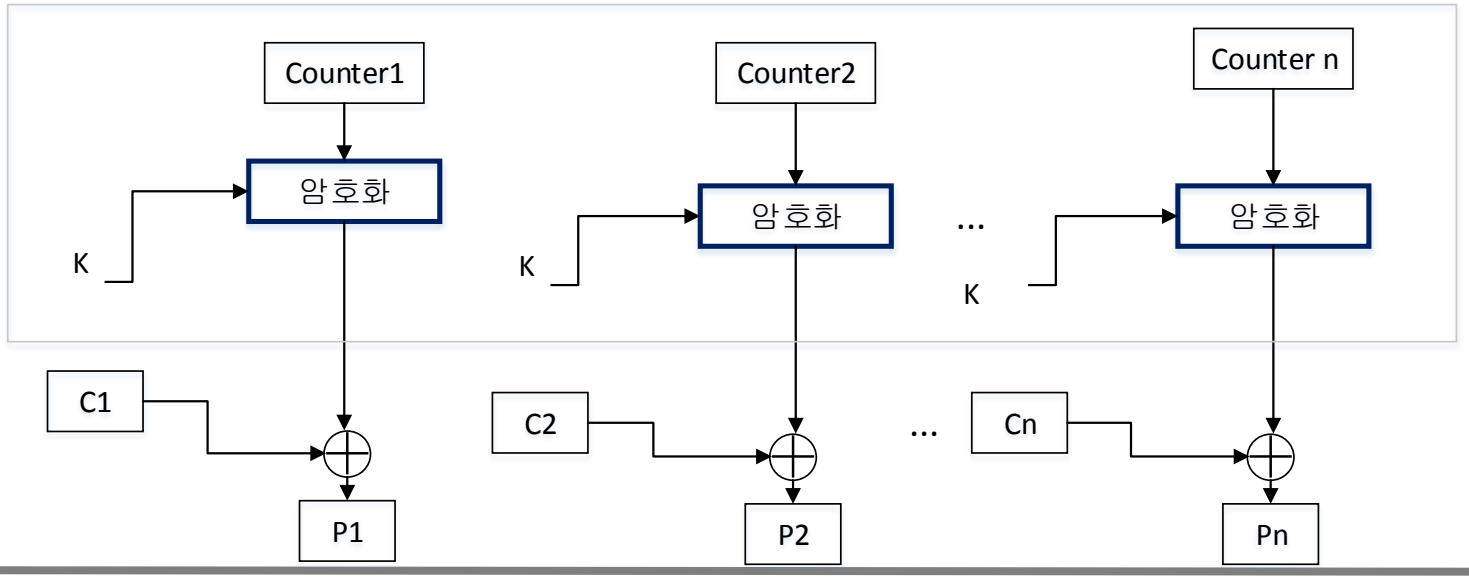
- CTR의 장단점
  - 장점
    - CBC와 달리 패딩 필요 없음CFB와 달리 오류 확산 없음
    - OFB와 달리 암호화, 복호화가 둘다 병렬처리 가능
    - CTR 모드의 암호화와 복호화가 완전히 같은 구조, 프로그램으로 구현하는 것이 매우 간단
    - 블록을 임의의 순서로 암호화· 복화화 할 수 있음
    - 카운터의 초기값이 실행 할 때 마다 다른 값으로 입력되기 때문에 해독이 어려움

카운터 모드

(ㄱ) 암호화



(ㄴ) 복호화



암호 블록 운용 모드

- 암호 블록 운용 모드 비교 표

암호 운용 모드	병렬 처리	패딩	랜덤 접근	초기화 백터	오류 확산
ECB	O	O	암호화, 복호화	X	X
CBC	복호화만	O	복호화	O	E: 해당 블록 이후 모든 블록
CFB	복호화만	X	복호화	O	D: 해당 블록과 다음 블록
OFB	X	X	X	O	X
CTR	O	X	암호화, 복호화	X	X

감사합니다!