2018/01/09, 2018 보안 기초 세미나

Network Security Essentials

- Chapter_2 대칭 암호와 메시지 기밀성(2)-

김경선 (kyeongseon@pel.smuc.ac.kr)

상명대학교 프로토콜공학연구실

목 차

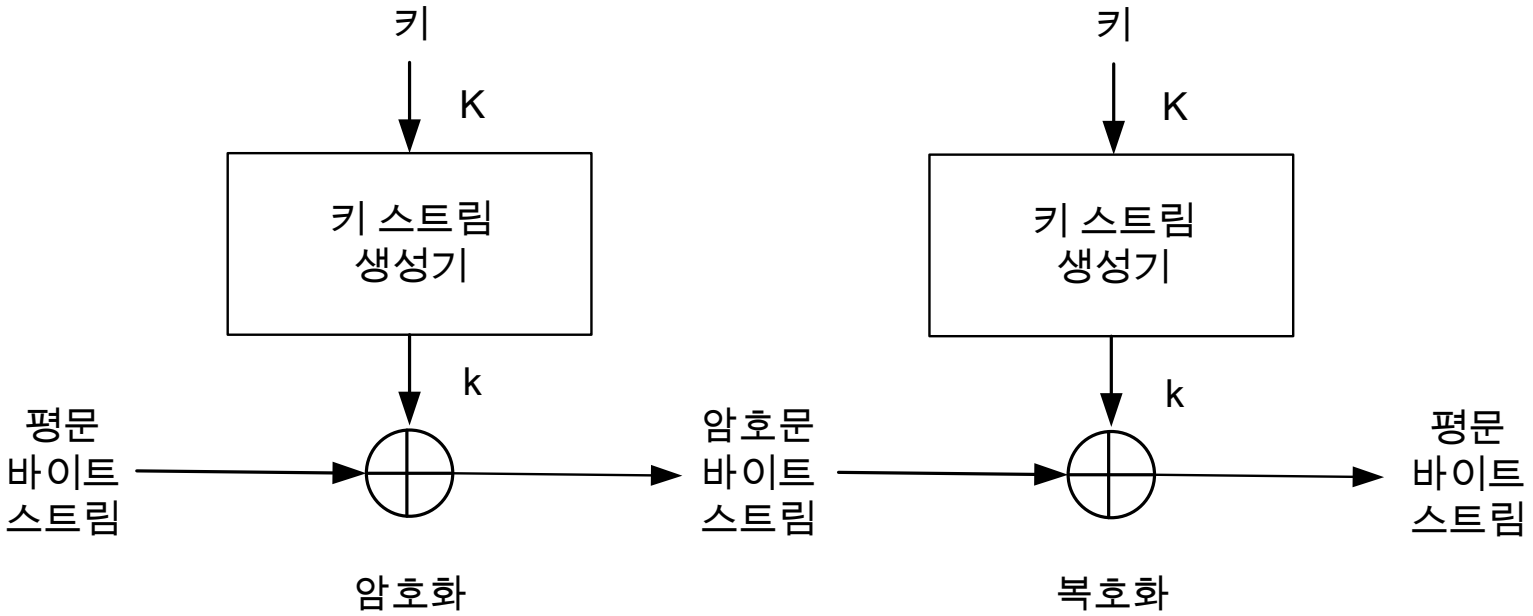
- 스트림 암호
  - RC4
- 암호 블록 운영 모드
  - 전자 코드북 모드
  - 암호 블록 체인 모드
  - 암호 피드백 모드
  - 출력 피드백 모드
  - 카운터 모드

스트림 암호

- 스트림 암호(Stream Cipher)
  - 한 번에 한 바이트씩 암호화하는 대칭키 암호 알고리즘
  - 장점
    - 속도가 빠름
    - 동등한 길이의 키를 사용하는 블록 암호만큼의 보안성을 유지
    - 블록 암호보다 작은 양의 코드로 구현
    - 스트림 암호화가 필요한 경우 적합
      - e.g., 음성, 영상 스트리밍
  - 단점
    - 2개의 평문을 동일한 키로 암호화 하는 경우 암호 해독이 아주 단순해짐
      - 두 개의 암호문 XOR은 원래의 평문과의 XOR과 같음

스트림 암호

- 구조



	11001100	평문
⊕	01101100	키스트림
	10100000	암호문

<암호화>

	10100000	암호문
⊕	01101100	키스트림
	11001100	평문

<복호화>

스트림 암호

- 속도

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

스트림 암호

- 스트림 암호 설계 주의사항

1. 주기가 긴 암호열
   - 의사 랜덤 넘버 생성기에서 무한 비트열을 생성하는데 주기가 길면 해독하기 더 어려워짐

2. 진성 랜덤 넘버 특성에 근사한 키스트림
   - 키스트림이 더 랜덤하게 구성되면 암호문도 더 랜덤해져 해독이 어려워짐

3. 충분히 긴 키의 길이
   - 전수공격을 차단

RC4

- RC4
  - 스트림 암호용 알고리즘 중 대표적인 알고리즘
  - 바이트 단위로 작동하는 다양한 크기의 키 사용
  - 랜덤 치환 기법 사용
  - 암호주기가 10^{100}보다 큼

- RC4 알고리즘 동작과정

1. S의 초기화
2. S의 초기 치환
3. 스트림 생성

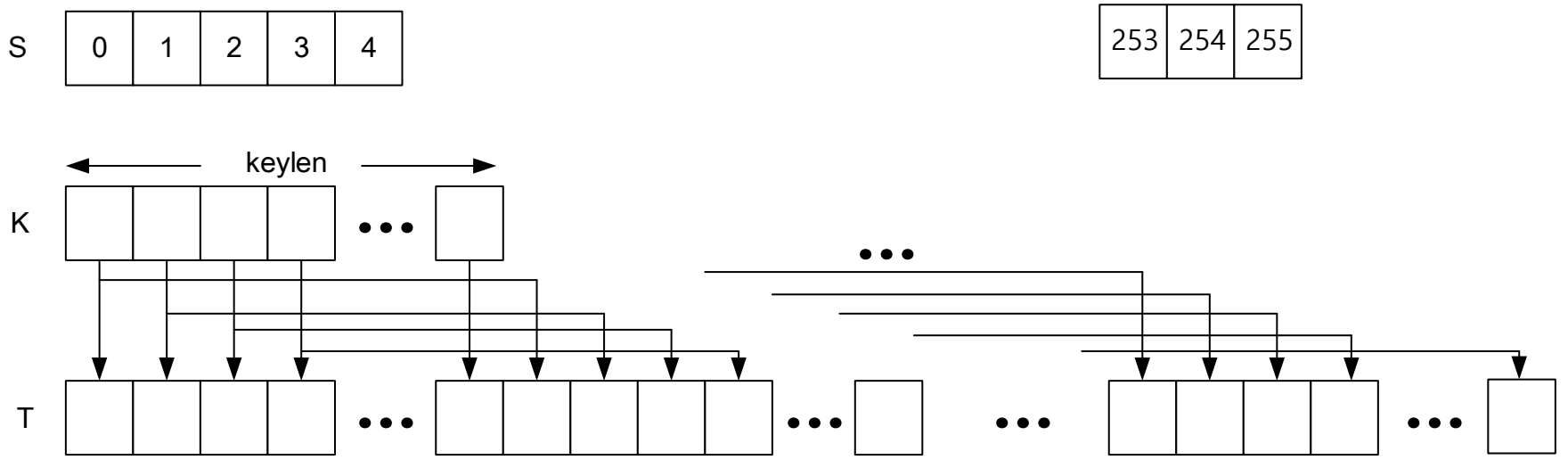
RC4

- RC4 알고리즘
  1. S의 초기화
     - 초기벡터 S 항목들을 오름차순으로 0부터 255까지의 값이 되도록 설정
       - S[0]=0, S[1]=1, …, S[256]=256
     - 임시벡터 T 생성
       - 키 값의 길이가 256바이트일 때
         - K를 그대로 T로 전달
       - 키 값의 길이가 256바이트 미만일 때
         - 키의 길이 Keylen만큼 T에 전달하고, T가 채워질 때까지 K를 반복하여 채움

RC4

- RC4 알고리즘

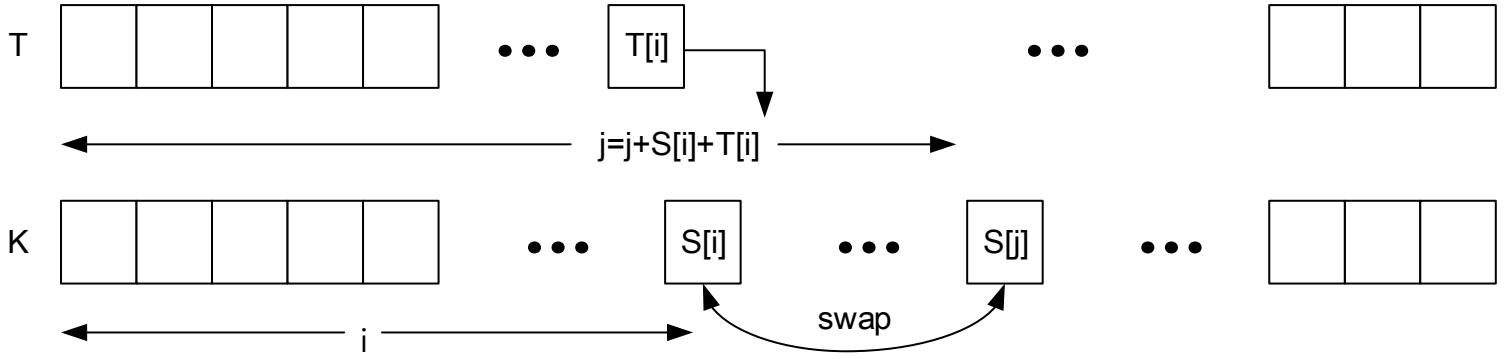
1. S의 초기화
   - RC4에서 S(상태 벡터) 초기화
     - /*Initialization*/
     - for i = 0 to 255 do
     - S[i] = i;
     - T[i] = K[i mod keylen];



RC4

- RC4 알고리즘

2. S의 초기 치환
   - S[0]부터 시작하여 S[255]까지 전체 배열 치환 반복 수행
   - 초기 S[i]에 대하여 T[i] 구조를 적용하여 S[j]로 랜덤 치환
   - S에서는 치환만 일어남



RC4

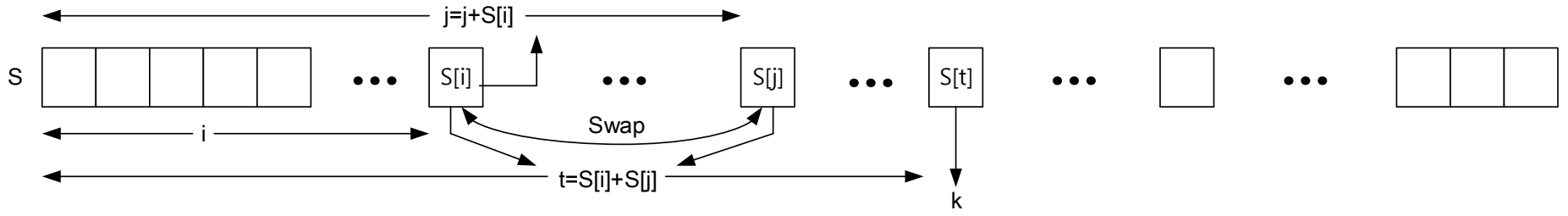
- RC4 알고리즘

2. RC4에서 T를 사용한 S 초기 치환
   - /*Initial Permutation of S*/
   - i = 0;
   - for i = 0 to 255 do
   - i = (i + S[i] + T[i]) mod 256;
   - Swap(S[i], S[j]);

RC4

- RC4 알고리즘

3. 스트림 생성
   - 랜덤한 t값을 찾아 해당 인덱스에 저장된 S배열에 각 k를 추출
   - K와 평문을 XOR



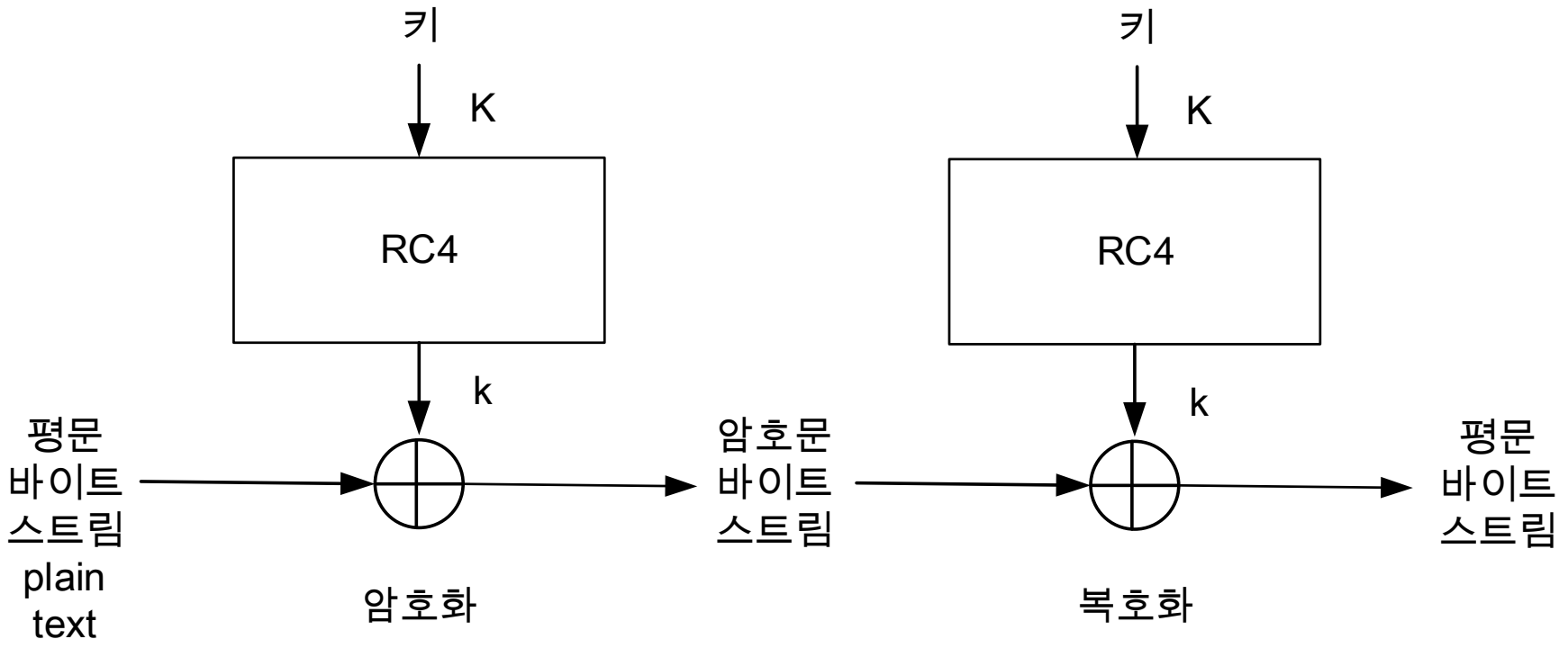
RC4

- RC4 알고리즘

3. 스트림 생성
   - RC4에서 스트림 생성
     - /*Stream Generation*/
     - i, j = 0;
     - while(true)
     - i = (i + 1) mod 256;
     - j = (j + S[i]) mod 256;
     - Swap(S[i], S[j]);
     - t = (S[i] + S[j]) mod 256;
     - k = S[t];
     - return plaintext XOR k

RC4

- RC4 전체 동작 과정
  - RC4 결과로 나온 키 스트림을 평문과 XOR해서 암호문을 만듬



암호 블록 운영 모드

- 암호 블록 운영 모드
  - 블록 암호를 사용할 때, 암호 알고리즘을 어떻게 사용하는가를 정하는 것
    - e.g., DES-CBC, AES-CTR
  - 평문의 길이가 블록 암호의 블록 크기보다 클 경우에 적용하여 해결

암호 블록 운영 모드

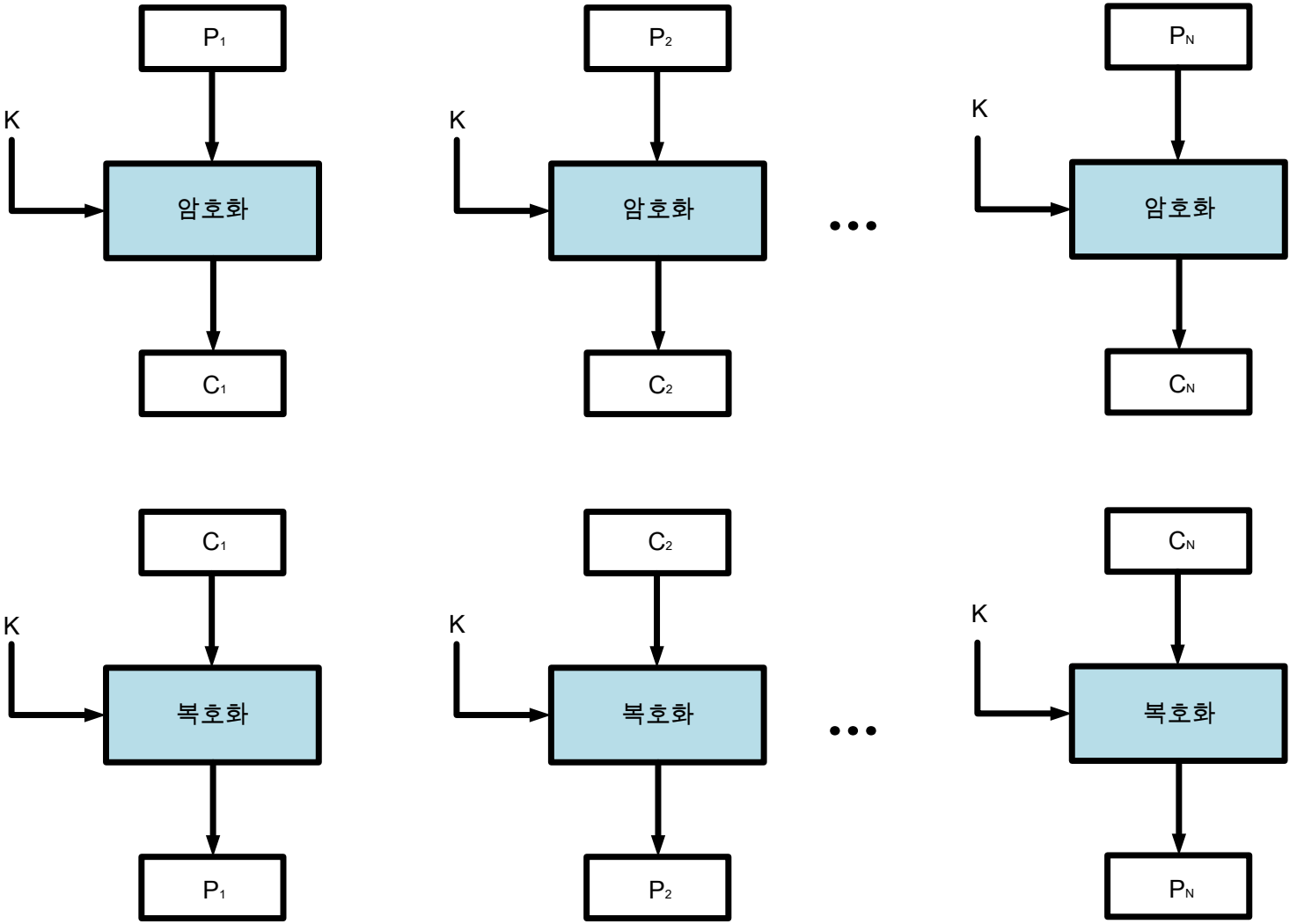
- 5가지 운용모드
  - 전자 코드북 모드(Electronic Codebook)
  - 암호 블록 체인 모드(Cipher Block Chaining)
  - 암호 피드백 모드(Cipher Feedback)
  - 출력 피드백 모드(Output Feedback)
  - 카운터 모드(Counter)

전자 코드북 모드

- 전자 코드북 모드(ECB)
  - 운영방식 중 가장 간단
  - 평문을 여러 블록으로 나누어 각각 암호화 하는 방식
  - 각 블록을 동일한 키로 암호화
    - 두 블록이 같은 값을 가진다면 암호화 한 결과가 같음
  - n비트 크기의 블록에 유일한 하나의 암호블록이 대응하므로 코드북(Codebook)이라 지칭

전자 코드북 모드

- 전자 코드북 모드(ECB)



전자 코드북 모드

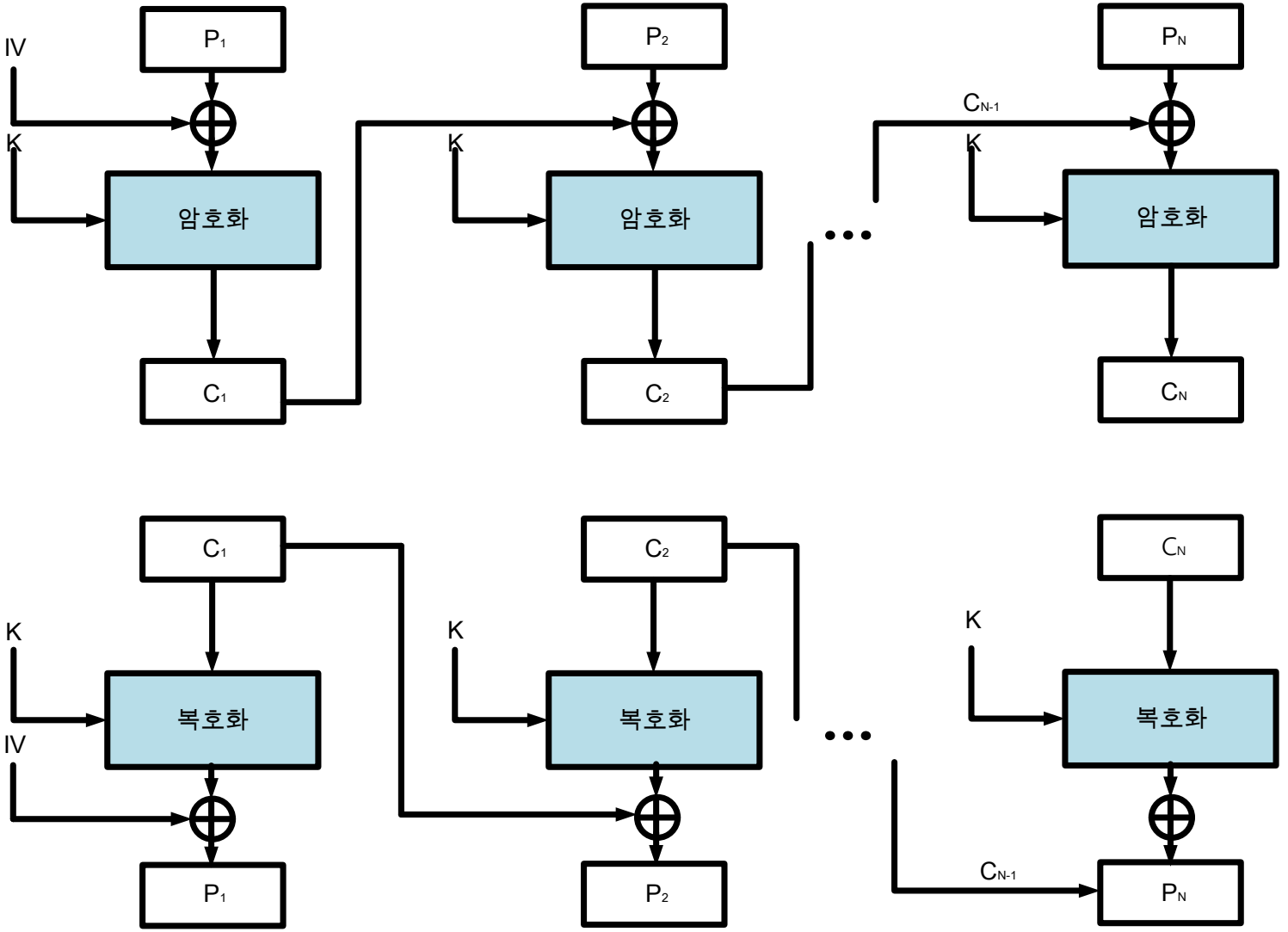
- 전자 코드북 모드(ECB)
  - 장점
    - 한 블록에서 에러가 나도 다른 블록에 영향을 주지 않음
      - 각 블록이 독립적으로 동작
    - 복호화 후 평문을 알기 위해 패딩(Padding)을 해야함
      - 암호문이 블록의 배수
  - 단점
    - 평문의 반복 패턴이 암호문에 그대로 나타남
    - 보안에 취약함
      - 한 개의 블록만 해독되면 나머지 블록도 해독 됨

암호 블록 체인 모드

- 암호 블록 체인 모드(CBC)
  - 평문의 각 블록은 XOR연산을 통해 이전 암호문과 연산됨
  - 블록 암호화 운영 모드 중 보안성이 가장 높은 암호화 방법
    - 가장 많이 사용
  - 초기화 벡터(Initial Vector) 사용
    - 첫번째 암호문에 대해서는 IV(Initial Vector)를 암호문 대신 사용
    - 송신자와 수신자가 모두 알고 있어야함
    - 보안성 강화를 위해 IV로 보낼 때 ECB 암호를 이용하여 보냄
    - 공격자가 유사한 IV로 수신자를 속일 수 있음

암호 블록 체인 모드

- 암호 블록 체인 모드(CBC)

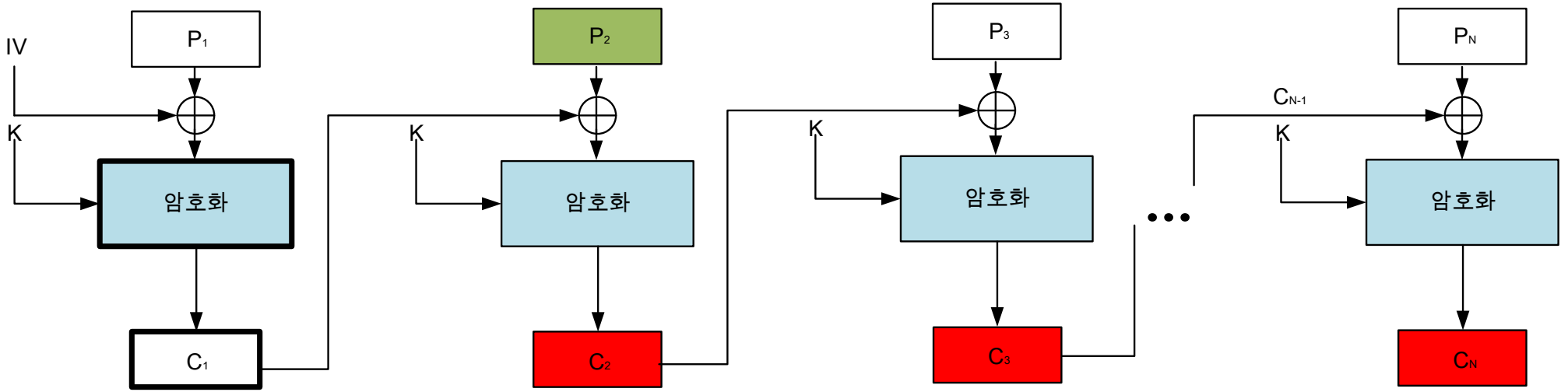


암호 블록 체인 모드

- 암호 블록 체인 모드(CBC)
  - 장점
    - 반복 패턴이 나타나지 않음
  - 단점
    - 평문을 알기 위해 패딩을 해야함
      - 암호문이 블록의 배수가 되기 때문
    - 암호화가 병렬처리가 아닌 순차적으로 수행되어야 함
    - 에러가 나면 깨진 암호문의 해당블록과 다음블록의 평문까지 영향을 미치게 됨

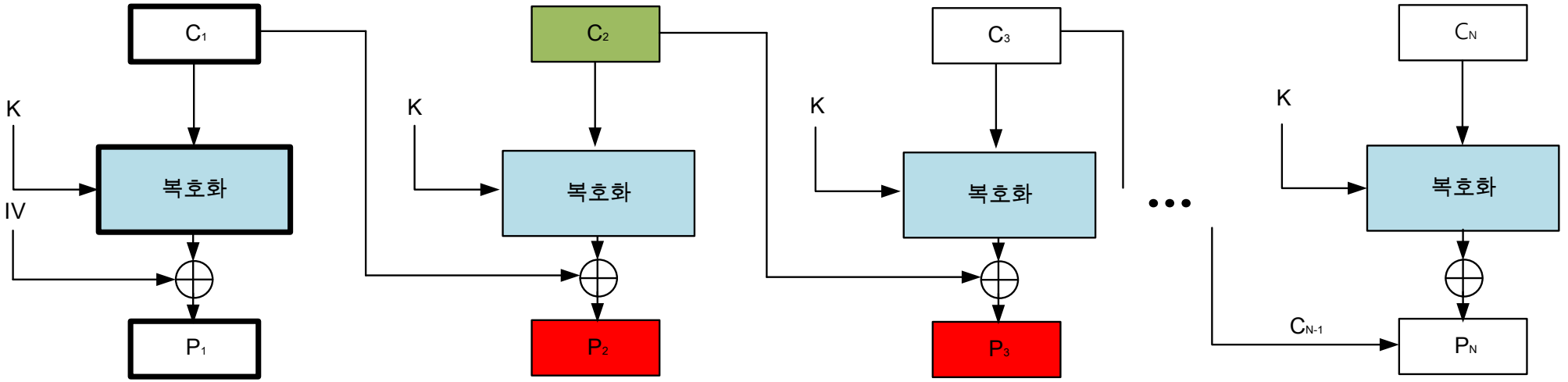
암호 블록 체인 모드

- 암호 블록 체인 모드(CBC)
  - 오류 확산(1/2)
    - 암호화 과정
      - 평문 블록 하나가 오류 났을 때, 현재 암호문 블록 이후 전체 암호문 블록에 영향



암호 블록 체인 모드

- 암호 블록 체인 모드(CBC)
  - 오류 확산(2/2)
    - 복호화 과정
      - 암호문 블록 하나가 오류 났을 때, 현재 평문 블록과 다음 평문 블록에만 영향

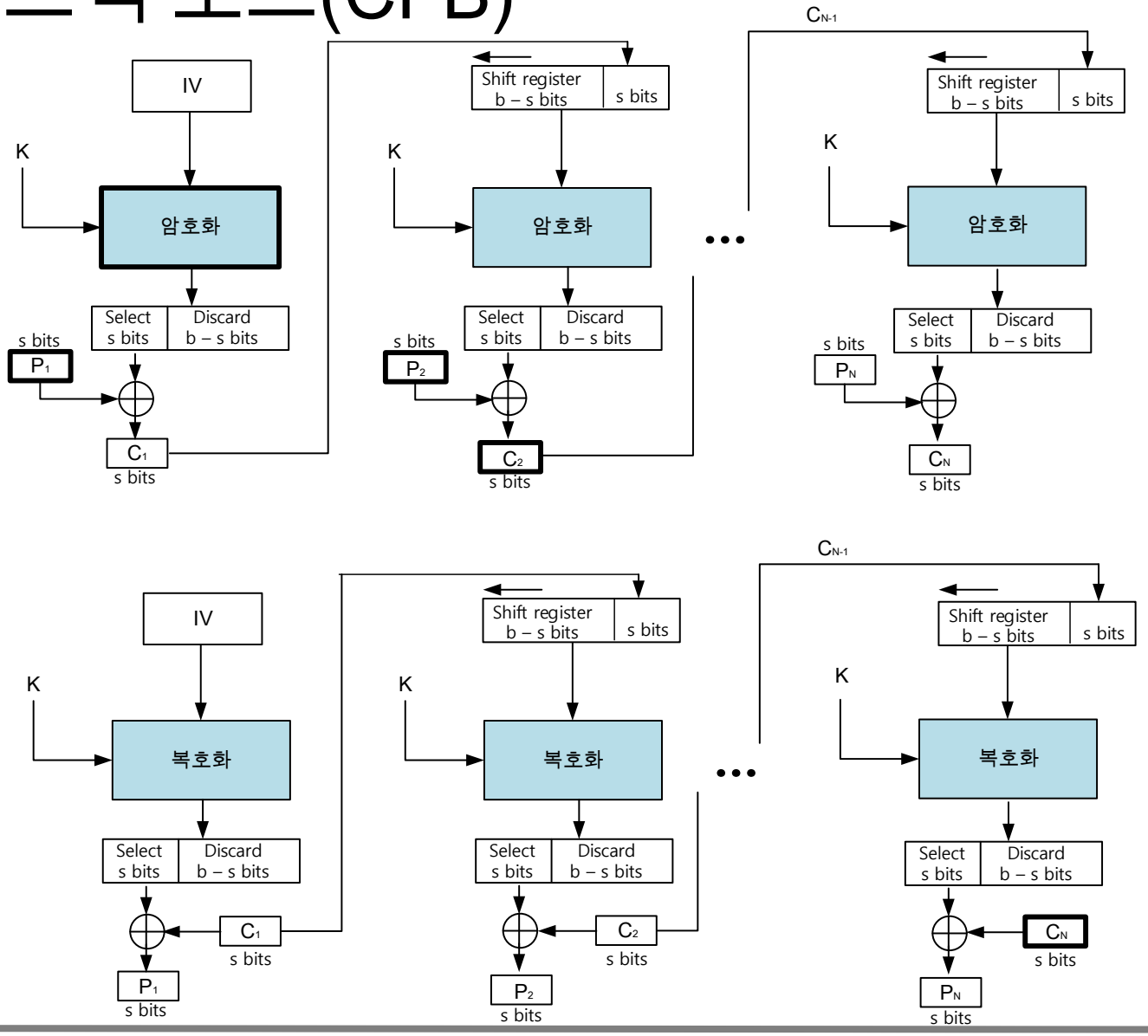


암호 피드백 모드

- 암호 피드백 모드(CFB)
  - CBC를 변형해 블록 암호를 스트림 암호로 변환
  - 암호문의 길이가 평문의 길이와 같도록 하는 특성

암호 피드백 모드

- 암호 피드백 모드(CFB)

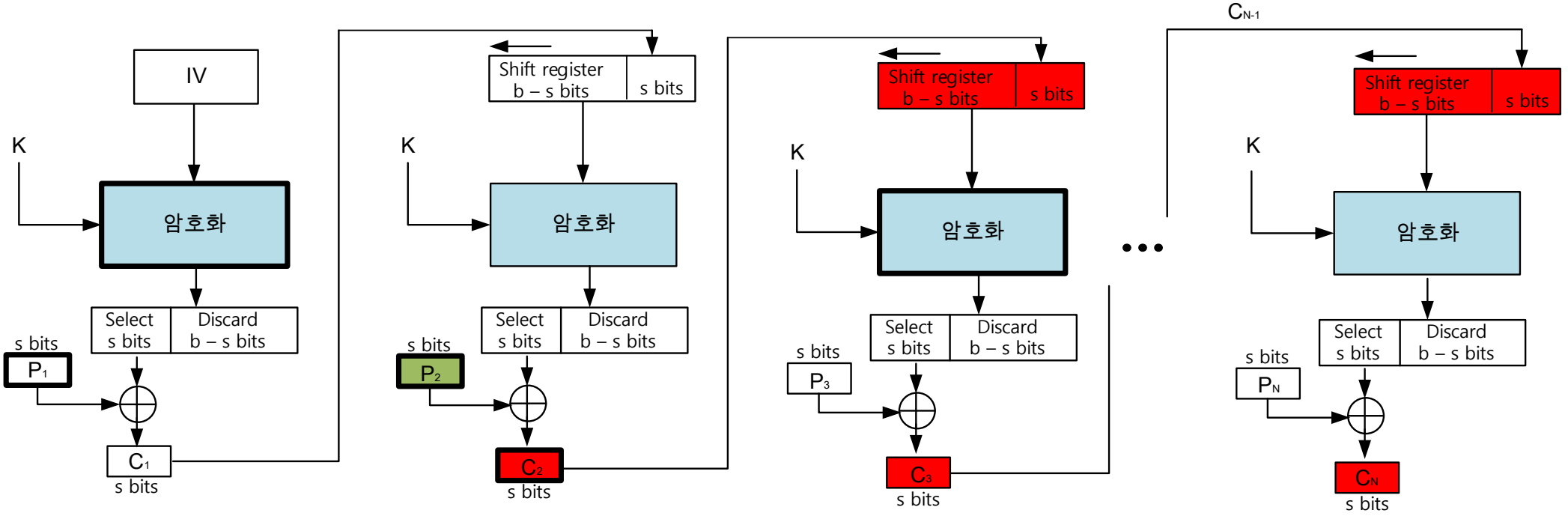


암호 피드백 모드

- 암호 피드백 모드(CFB)
  - 장점
    - 패딩(Padding)이 필요 없음
      - 암호문의 길이와 평문의 길이가 같기 때문
    - 복호화가 병렬적으로 처리 됨
  - 단점
    - 암호화, 복호화가 같은 방식
    - 에러가 나면 해당 평문 블록과 다음 평문 블록, 총 두 개의 블록에 전파됨
    - 암호화가 순차적으로 처리 됨

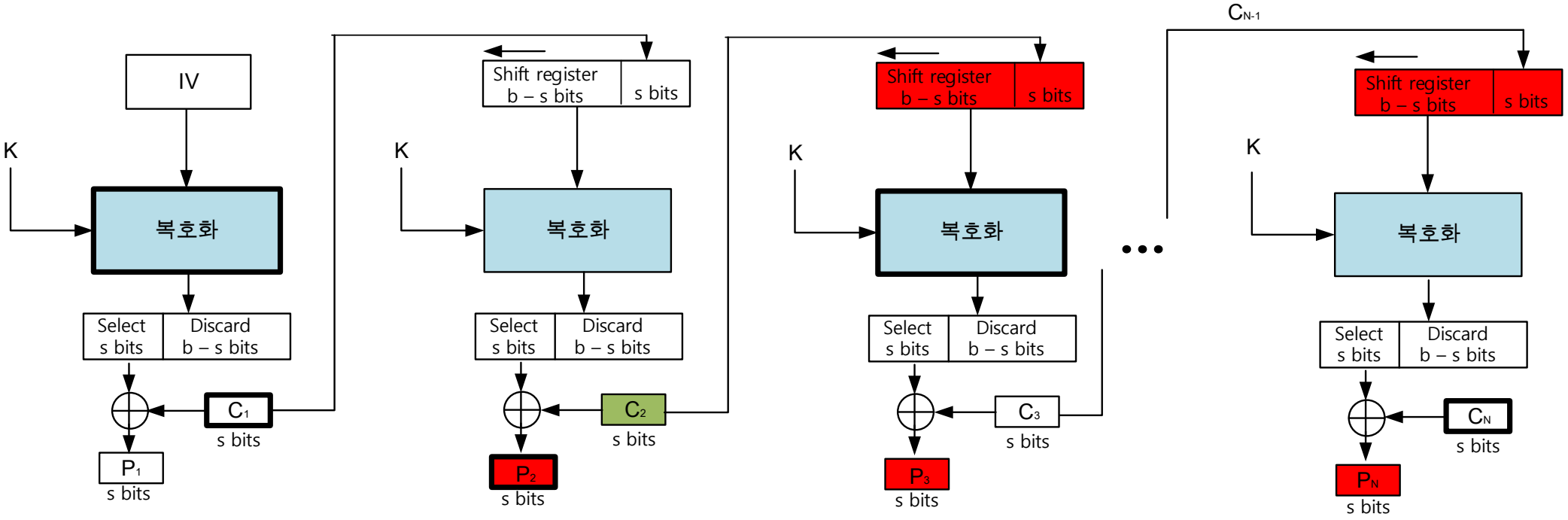
암호 피드백 모드

- 암호 피드백 모드(CFB)
  - 오류 확산(1/2)
    - 암호화 과정
      - 평문 블록 하나가 오류 났을 때, 현재 암호문 블록 이후 Shift register 에서 오류가 완전히 소멸될 때까지 암호문 블록에 영향



암호 피드백 모드

- 암호 피드백 모드(CFB)
  - 오류 확산(2/2)
    - 복호화 과정
      - 암호문 블록 하나가 오류 났을 때, 현재 평문 블록 이후 Shift register 에서 오류가 완전히 소멸될 때까지 평문 블록에 영향

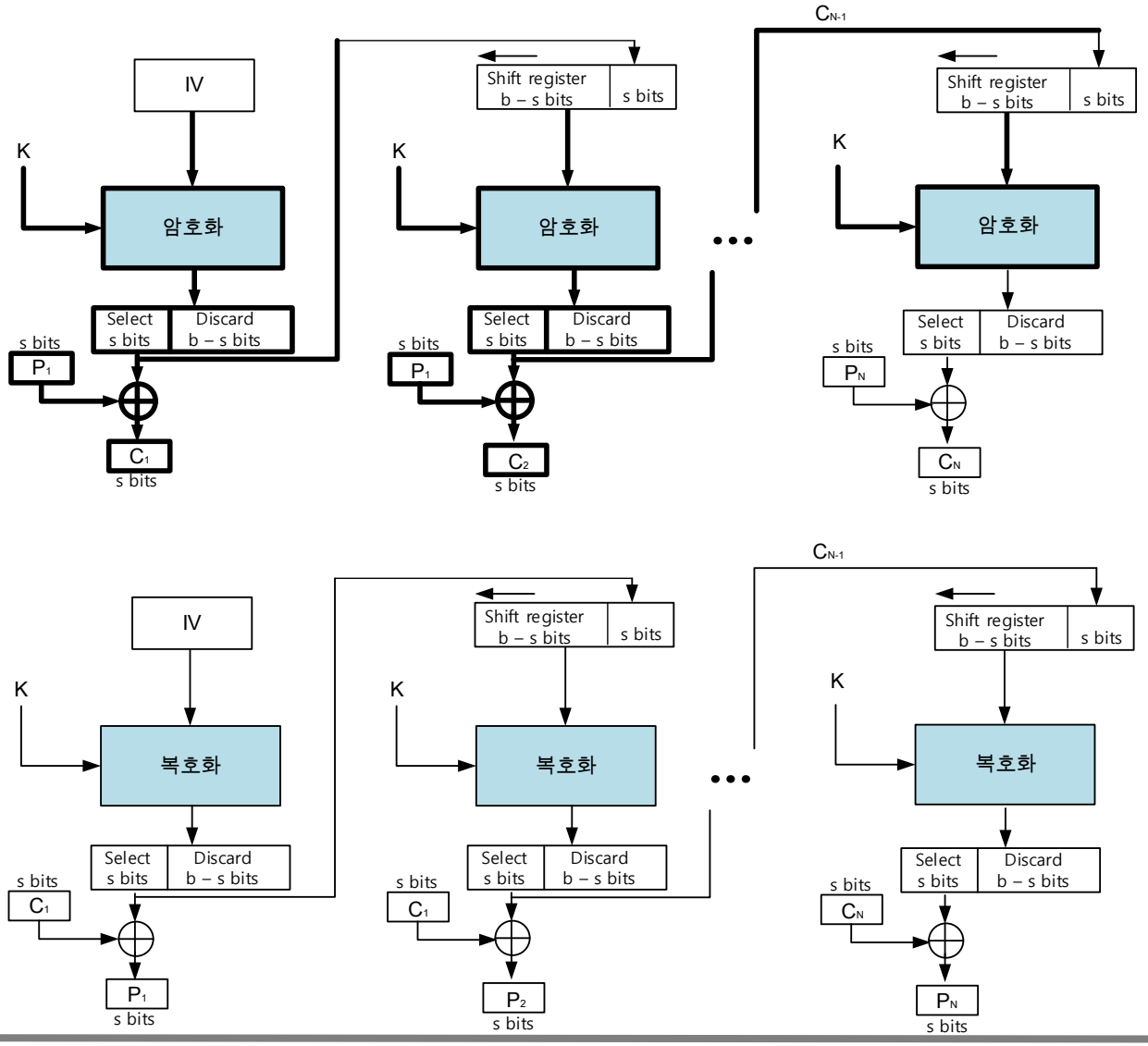


출력 피드백 모드

- 출력 피드백 모드(OFB)
  - 암호화와 암호 해제방식이 동일해 두 번 암호화 하면 평문이 나옴
  - 암호문의 길이가 평문의 길이와 같도록 하는 특성

출력 피드백 모드

- 출력 피드백 모드(OFB)

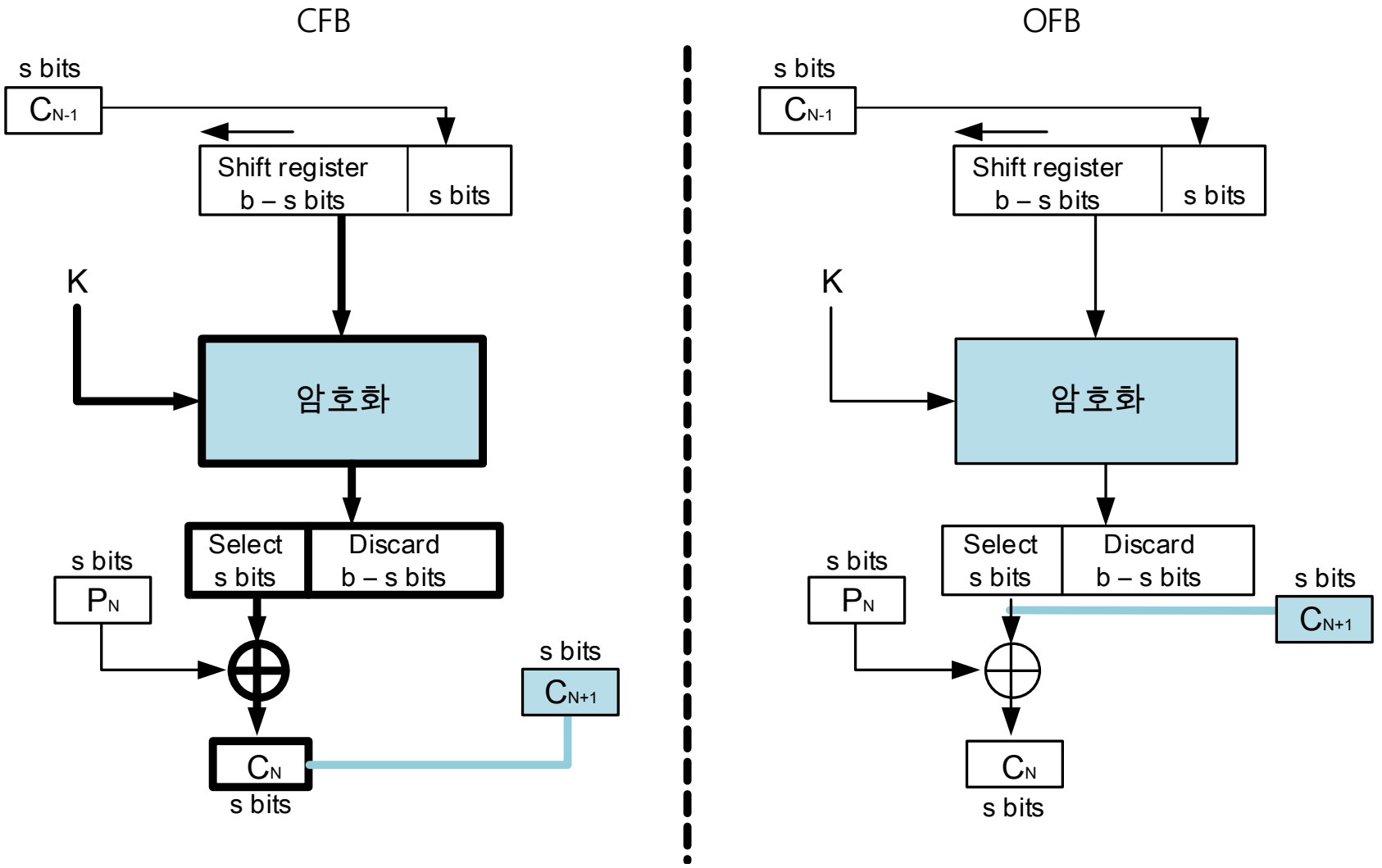


출력 피드백 모드

- 출력 피드백 모드(OFB)
  - 장점
    - 패딩이 필요 없음
      - 암호문의 길이와 평문의 길이가 같기 때문
    - 오류가 확산 되지 않음
  - 단점
    - 병렬처리 불가능

출력 피드백 모드

- 암호 피드백 모드와 출력 피드백 모드 비교



카운터 모드

- 카운터 모드(CTR)
  - 블록을 암호화 할 때마다 1씩 증가해가는 카운터를 암호화해서 키스트림을 만듬
  - 카운터를 암호화한 비트열과 평문 블록과의 XOR를 취한 결과가 암호문 블록이 됨

  - 카운터 만드는 법
    - 앞부분의 8바이트는 비표로 암호화 할 때마다 다른 값으로 함
    - 뒷부분 8바이트는 블록 번호로 카운트해서 하나씩 증가

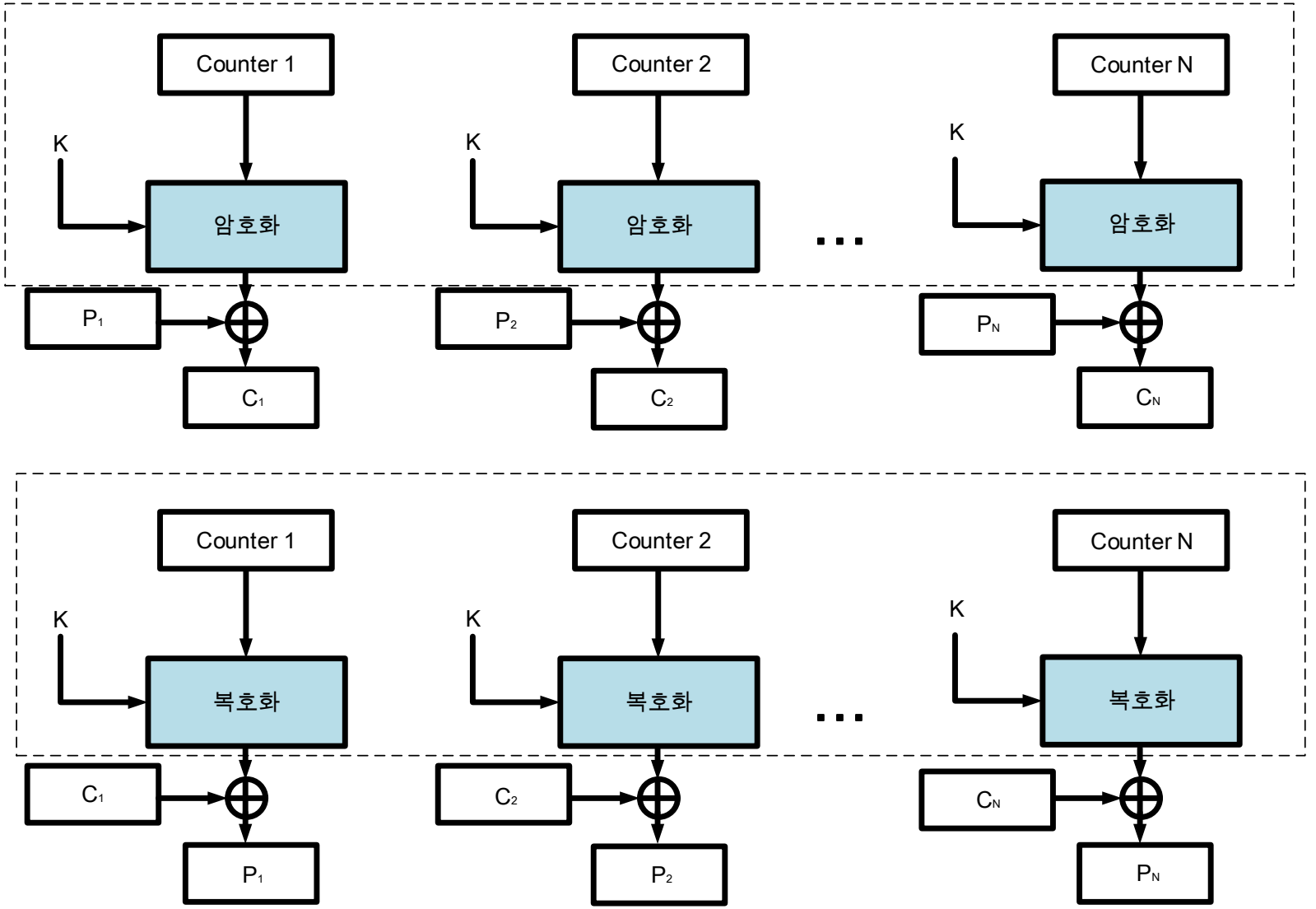
66 1F 98 CD 37 A3 8B 4B 00 00 00 00 00 00 00 01

비표(Nonce) 블록 번호

<블록 길이가 128비트(16바이트)인 경우 카운터 초기 값의 예>

카운터 모드

- 카운터 모드(CTR)

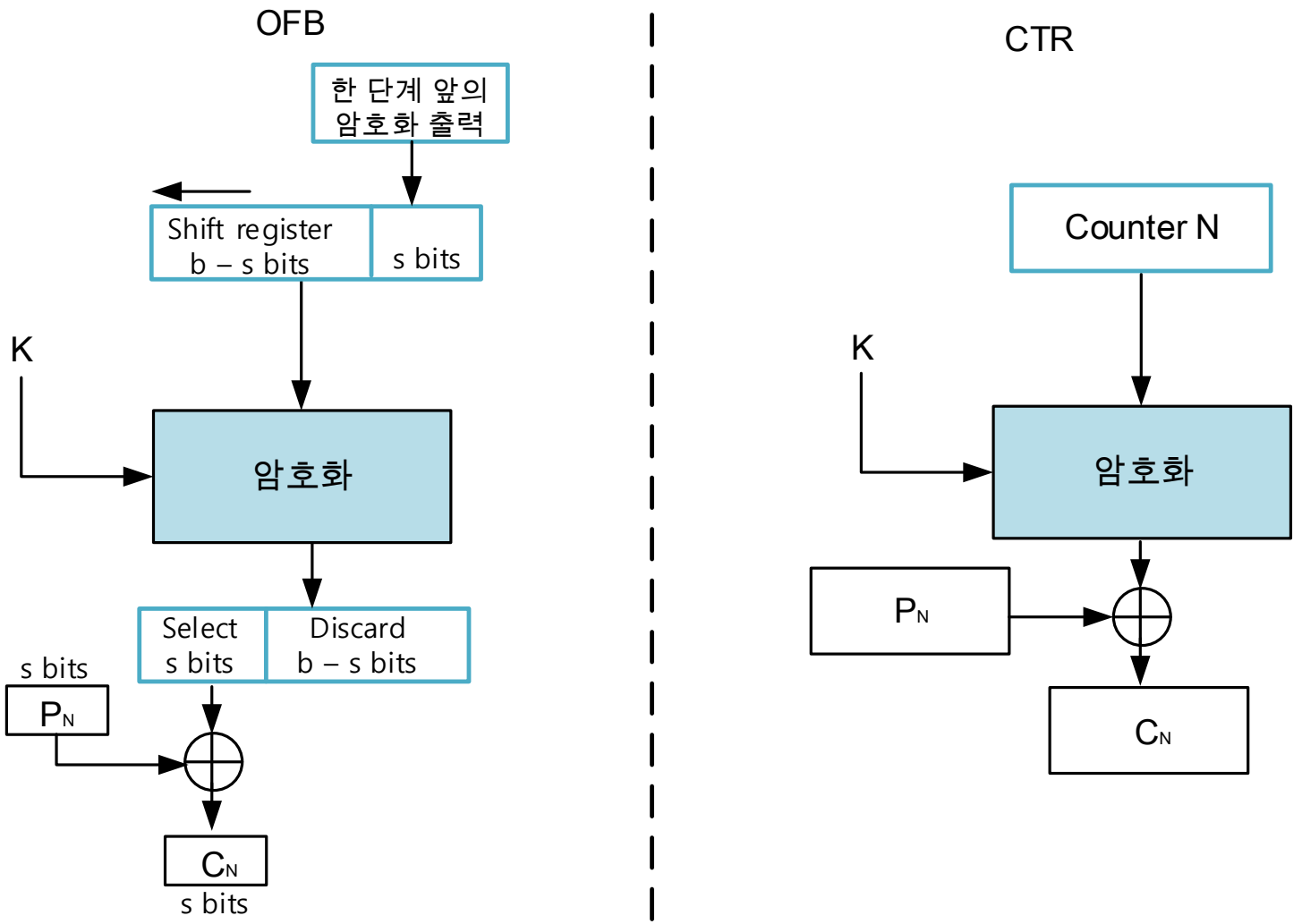


카운터 모드

- 카운터 모드(CTR)
  - 장점
    - 병렬 처리 가능
    - 오류가 확산되지 않음
    - 패딩이 필요하지 않음
  - 단점
    - 공격자가 암호문 블록을 비트 반전시키면, 대응하는 평문 블록이 비트 반전

카운터 모드

- 출력 피드백 모드와 카운터 모드 비교



암호 블록 운영 모드

- 블록 암호 운영 모드 비교 표

암호 운영 모드	병렬 처리	패딩	랜덤 접근	초기화 벡터	오류 확산	VI 사용
ECB	O	O	O	X	X	X
CBC	복호화만	O	복호화만	O	E: 해당 블록 이후 모든 블록 D: 해당 블록과 다음 블록	X
CFB	복호화만	X	복호화만	O	Shift register에서 오류가 완전히 소멸 될 때까지	O
OFB	X	X	X	O	X	O
CTR	O	X	O	X	X	x

감사합니다!