2023/01/18, 2023 겨울방학 보안기초 세미나

Network Security Essentials

- Chapter_2 대칭 암호와 메시지 기밀성(2) -

손 우 영(wooyoung@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

- 스트림 암호와 RC4
- 암호 블록 운용 모드

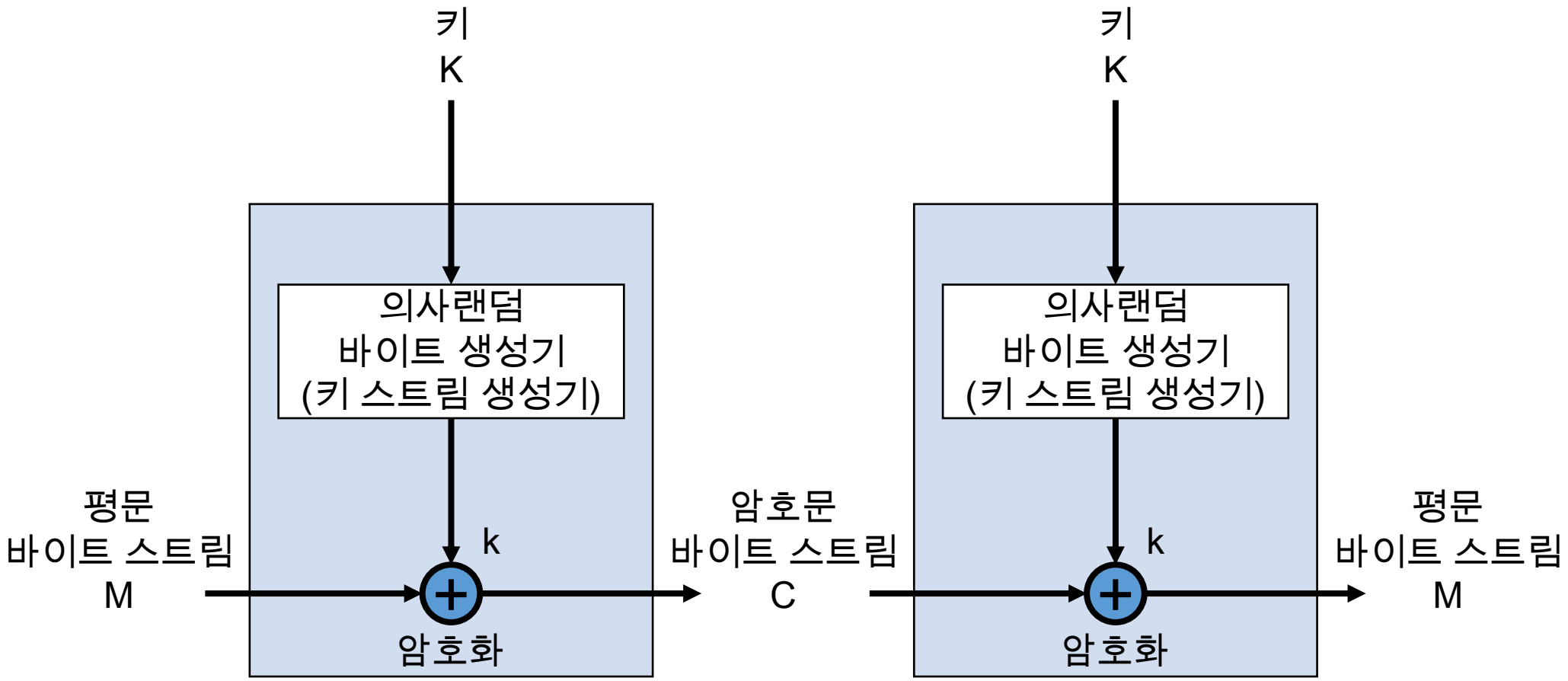
목 차

- 스트림 암호와 RC4
- 암호 블록 운용 모드

스트림 암호와 RC4

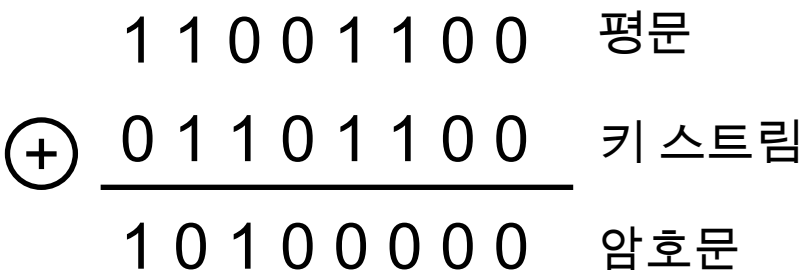
- 스트림 암호(Stream Cipher)
  - 정의
    - 입력되는 요소를 연속적으로 처리하는 대칭키 암호 구조
  - 구조
    - 의사난수 비트 생성기에 키를 입력하여 키 스트림이 출력
    - 이진 평문 스트림과 이진 키 스트림의 XOR 연산으로 암호문 생성
    - 암호화에 사용된 키 스트림과 암호문을 XOR 연산하여 복호화

스트림 암호와 RC4

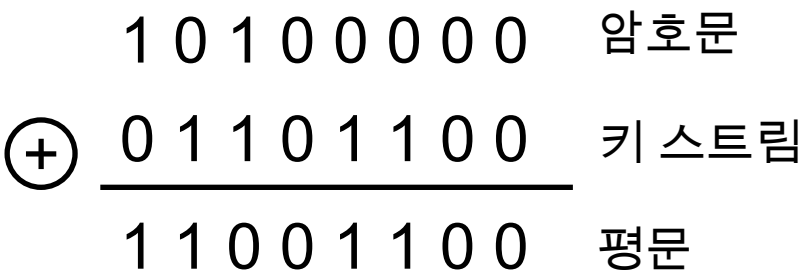

키
K
의사랜덤
바이트 생성기
(키 스트림 생성기)
k
평문
바이트 스트림
M
암호화
암호문
바이트 스트림
C
키
K
의사랜덤
바이트 생성기
(키 스트림 생성기)
k
암호화
평문
바이트 스트림
M

스트림 암호와 RC4

- 스트림 암호(Stream Cipher)
  - e.g., 키 스트림을 이용한 XOR 연산



<암호화>



<복호화>

스트림 암호와 RC4

- 스트림 암호(Stream Cipher)
  - 설계 시 고려사항
    - 암호열의 주기가 커야 함
    - 키 스트림은 진성 난수 스트림의 특성에 근사해야 함
    - 키의 길이가 충분히 길어야 함

스트림 암호와 RC4

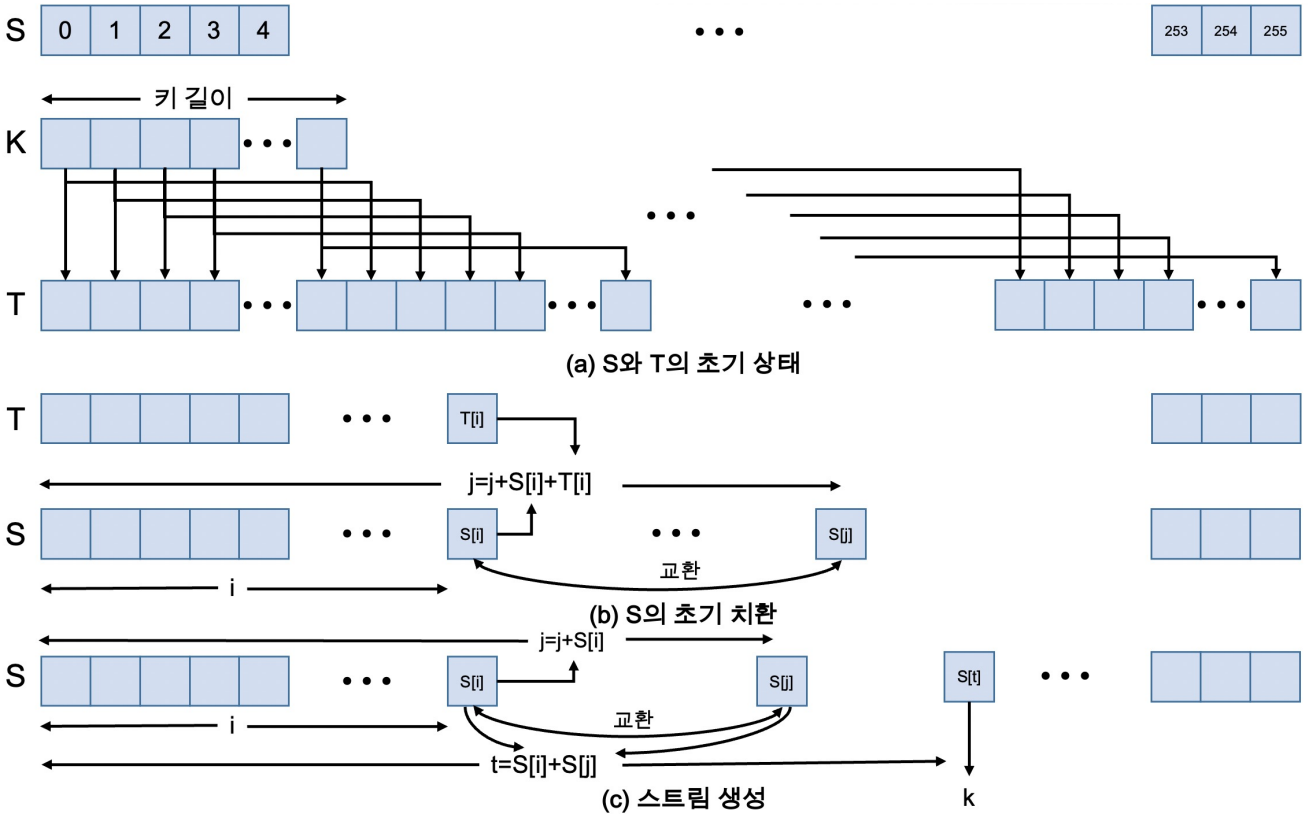
- 스트림 암호(Stream Cipher)
  - 장점
    - 블록 암호보다 속도가 빠름
    - 블록 암호와 달리 실시간 처리 가능
  - 단점
    - 두 개 이상의 평문을 동일한 키를 사용해서 암호화 한다면 암호 해독이 단순해짐

스트림 암호와 RC4

- 스트림 암호(Stream Cipher)
  - RC4 알고리즘
    - 정의
      - 바이트 단위로 작동되도록 만들어진 다양한 크기의 키를 사용하는 스트림 암호
    - 특징
      - 사용되는 알고리즘은 랜덤 치환에 기초하여 만들어짐
      - 빠르게 작동되는 암호화 알고리즘
      - WEP 프로토콜과 WPA 프로토콜에서 사용

스트림 암호와 RC4

- 스트림 암호(Stream Cipher)
  - RC4 알고리즘
    - 과정
      1. S와 T의 초기화
      2. S의 초기 치환
      3. 스트림 생성

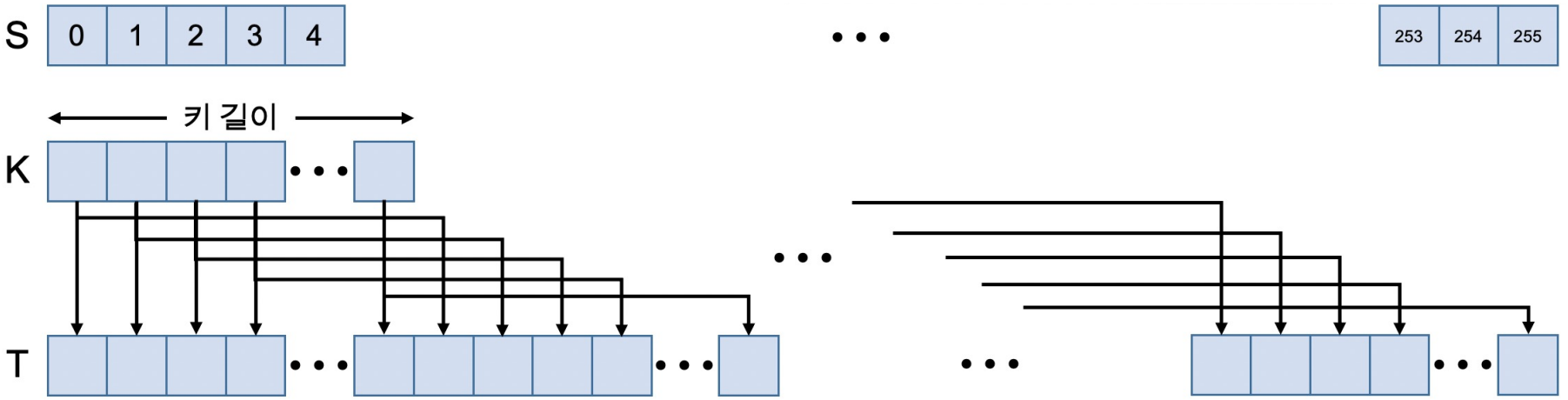


스트림 암호와 RC4

- 스트림 암호(Stream Cipher)
  - RC4 알고리즘
    - 과정(1/3)
      - S와 T의 초기화
        - S는 0부터 255까지 오름차순 정렬
        - T가 채워질 때까지 K값 저장

```
/*Initialization*/
for i = 0 to 255 do
    S[i] = i;
    T[i] = K[i mod keylen];
```

T[i] = K[i % strlen(K)]

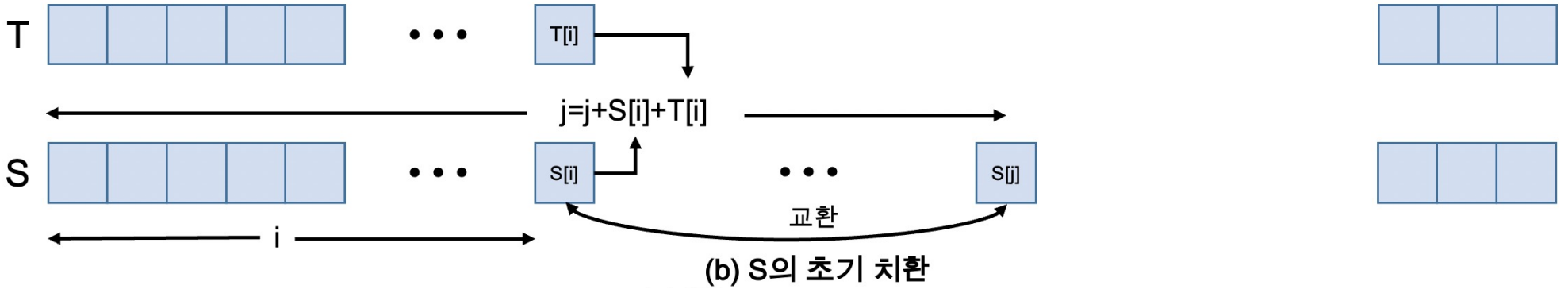


(a) S와 T의 초기 상태

스트림 암호와 RC4

- 스트림 암호(Stream Cipher)
  - RC4 알고리즘
    - 과정(2/3)
      - S의 초기 치환
        - T[i]를 이용한 구조에 따라 S[i]를 S의 다른 바이트와 교환

```
/*Initial Permutation of S*/
j = 0;
for i = 0 to 255 do
    j = (j + S[i] + T[i]) mod 256;
    Swap(S[i], S[j]);
```

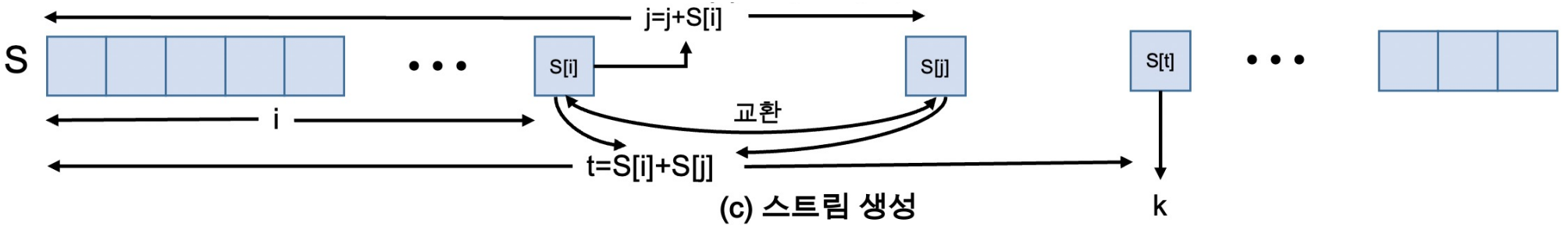


(b) S의 초기 치환

스트림 암호와 RC4

- 스트림 암호(Stream Cipher)
  - RC4 알고리즘
    - 과정(3/3)
      - 스트림 생성
        - i 값에 따른 j 값 계산
        - S[i]와 S[j]의 위치 교환
        - t 값에 따른 키 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];
```



(c) 스트림 생성

목 차

- 스트림 암호와 RC4
- 암호 블록 운용 모드

암호 블록 운용 모드

- 정의
  - 하나의 키를 사용하여 블록 암호를 반복적으로 이용하는 절차
- 종류
  - 전자 코드북 모드(ECB: Electronic Codebook Mode)
  - 암호 블록 체인 모드(CBC: Cipher Block Chaining Mode)
  - 암호 피드백 모드(CFB: Cipher Feedback Mode)
  - 카운터 모드(CTR: Counter Mode)

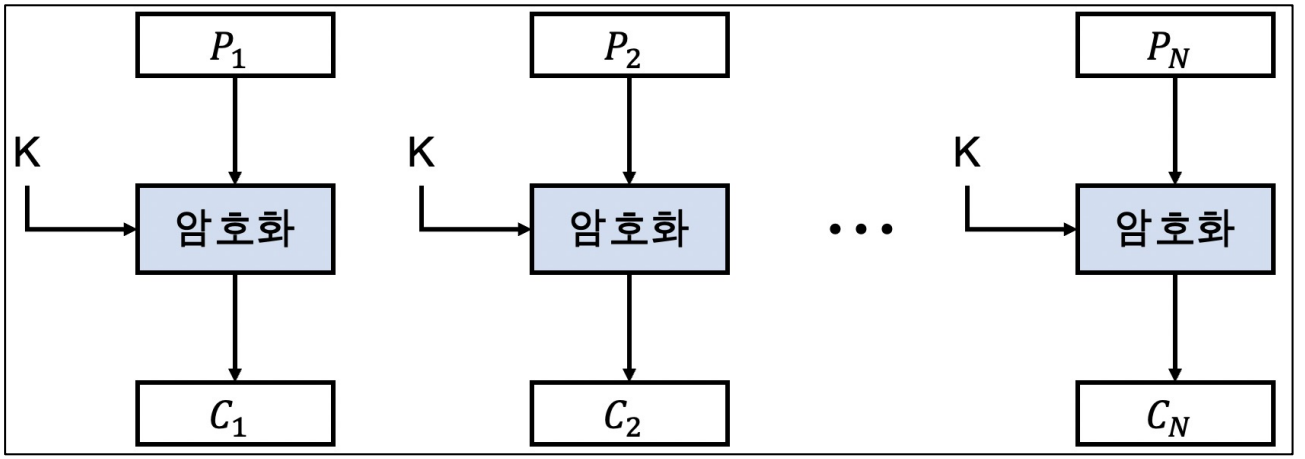
암호 블록 운용 모드

- 종류(1/4)
  - 전자 코드북 모드(ECB: Electronic Codebook Mode)
    - 정의
      - 평문을 일정한 크기의 블록으로 나누고 각 블록을 동일한 키로 암호화하는 방식
    - 특징
      - 운용 모드 중에서 가장 간단한 모드
      - 패딩이 필요함

암호 블록 운용 모드

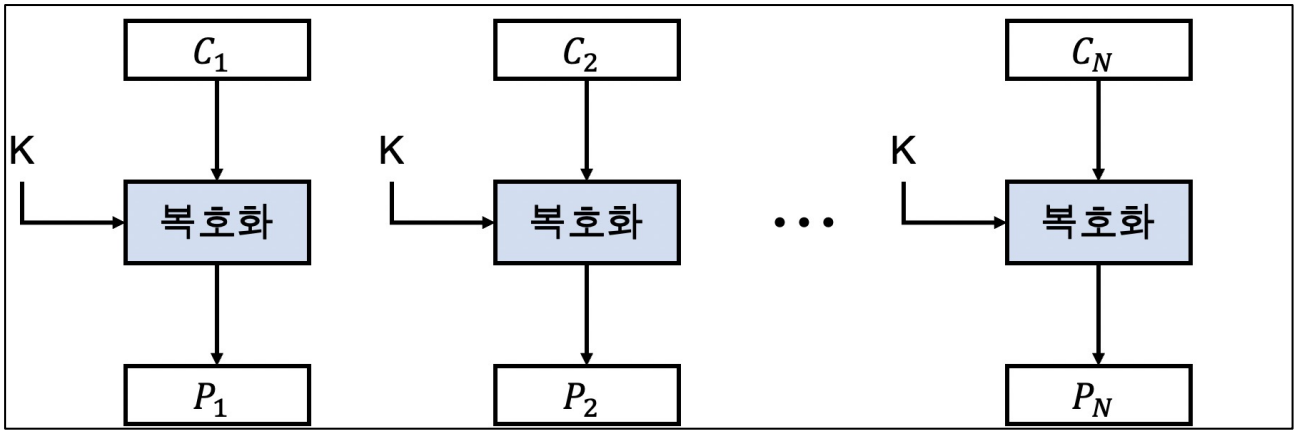
- 종류(1/4)
  - 전자 코드북 모드(ECB: Electronic Codebook Mode)
    - 암호화

$$C_i = E(K, P_i)$$



    - 복호화

$$P_i = D(K, C_i)$$

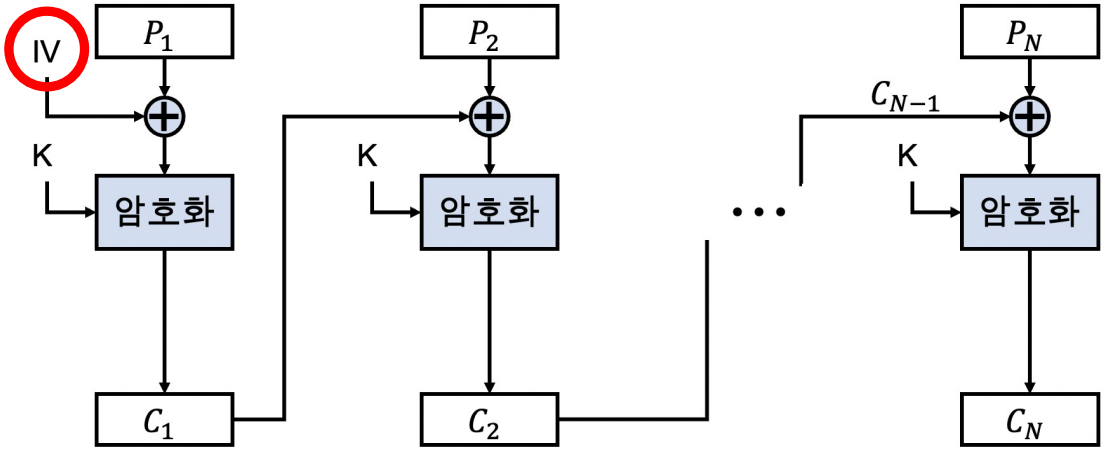


암호 블록 운용 모드

- 종류(1/4)
  - 전자 코드북 모드(ECB: Electronic Codebook Mode)
    - 안전성 문제
      - 블록 단위의 패턴이 유지됨
        - 평문에서 같은 값을 갖는 블록은 대응되는 암호문 블록도 같은 값을 가짐
      - 블록간의 독립성은 키를 알지 못해도 특정 암호문을 변조할 수 있는 기회 제공
    - 응용
      - 암호문 블록의 독립성
        - 병렬처리 가능

암호 블록 운용 모드

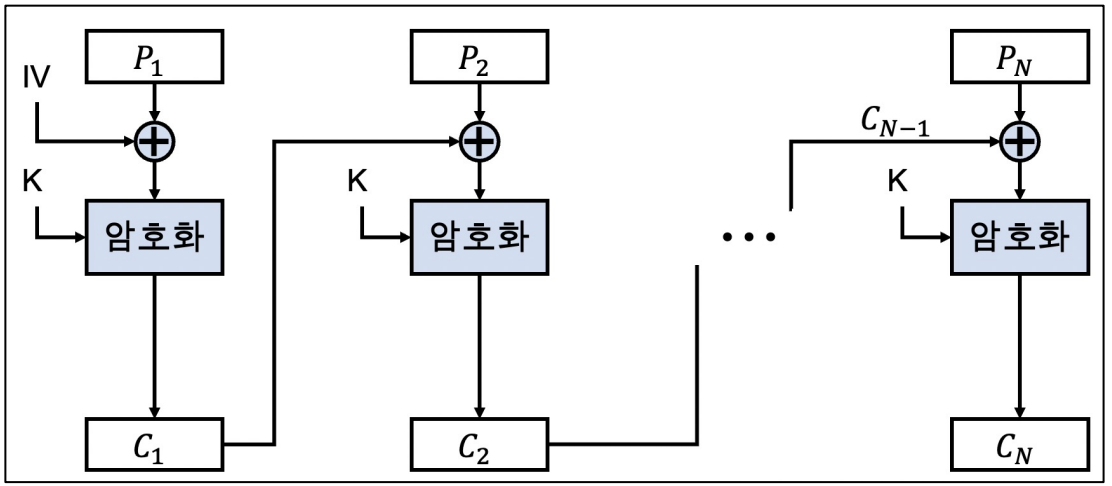
- 종류(2/4)
  - 암호 블록 체인 모드(CBC: Cipher Block Chaining Mode)
    - 정의
      - 체인 구조를 이루며 각 블록이 이전의 암호화 블록의 영향을 받는 방식
    - 특징
      - 초기화 벡터(IV: Initialization Vector) 사용



암호 블록 운용 모드

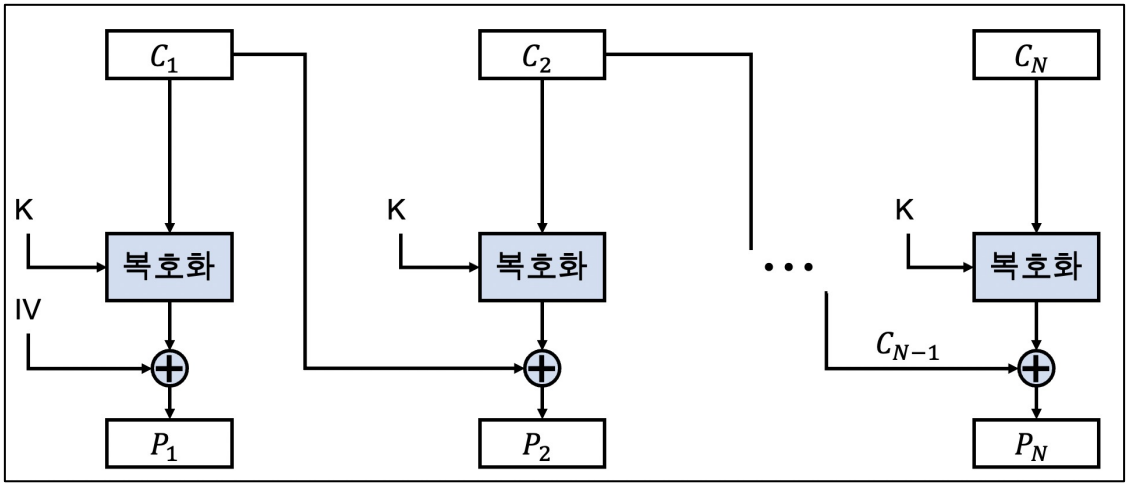
- 종류(2/4)
  - 암호 블록 체인 모드(CBC: Cipher Block Chaining Mode)
    - 암호화

$$C_i = E(K, [C_{i-1} \oplus P_i])$$



    - 복호화

$$P_i = D(K, C_i) \oplus C_{i-1}$$



암호 블록 운용 모드

- 종류(2/4)
  - 암호 블록 체인 모드(CBC: Cipher Block Chaining Mode)
    - 안전성 문제
      - 한 메시지내에서 블록 단위의 패턴이 유지되지 않음
        - 동일한 평문 블록들은 서로 다른 암호화 블록으로 암호화 됨
      - 동일한 초기 벡터를 사용하는 환경에서 두 개의 메시지가 동일하다면 대응되는 암호문은 동일함
    - 응용
      - 블록 간의 연관성 존재
        - 병렬 처리 불가능
        - 랜덤하게 선택된 파일을 암호화하거나 복호화할 때 사용 불가능

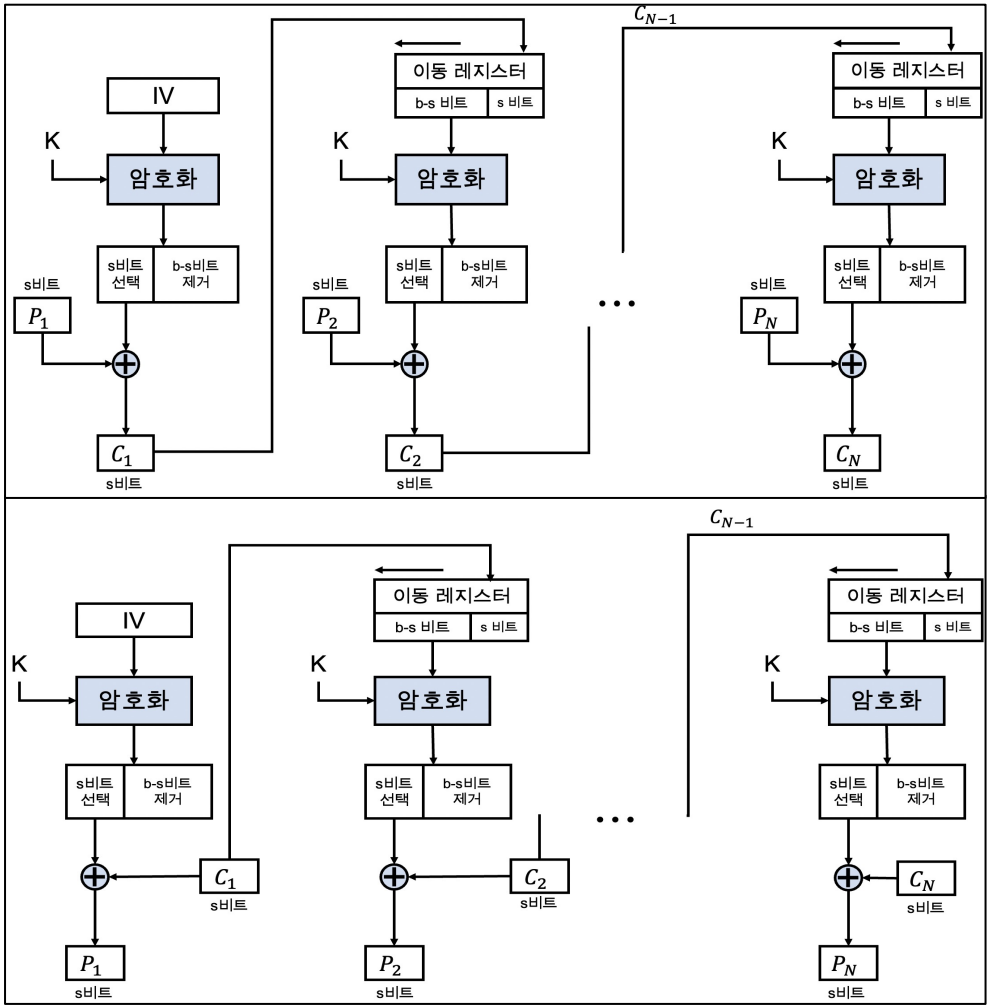
암호 블록 운용 모드

- 종류(3/4)
  - 암호 피드백 모드(CFB: Cipher Feedback Mode)
    - 정의
      - 이전 암호문 블록을 암호 알고리즘의 입력으로 사용하는 방식
    - 특징
      - 초기화 벡터(IV: Initialization Vector) 사용
      - 암호 피드백 모드를 이용하면 블록 암호를 스트림 암호를 바꿀 수 있음
        - 각 문자가 암호화 되는 즉시 전송 가능

암호 블록 운용 모드

- 종류(3/4)
  - 암호 피드백 모드(CFB: Cipher Feedback Mode)
    - 암호화

$$C_1 = P_1 \oplus S_S[E(K, IV)]$$



    - 복호화

$$P_1 = C_1 \oplus S_S[E(K, IV)]$$

암호 블록 운용 모드

- 종류(3/4)
  - 암호 피드백 모드(CFB: Cipher Feedback Mode)
    - 안전성 문제
      - 한 메시지내에서 블록 단위의 패턴이 유지되지 않음
        - 동일한 평문 블록들은 서로 다른 암호화 블록으로 암호화 됨
      - 동일한 초기 벡터를 사용하는 환경에서 두 개의 메시지가 동일하다면 대응되는 암호문은 동일함
    - 응용
      - 한 문자나 한 비트와 같은 작은 크기의 블록을 암호화하는데 사용

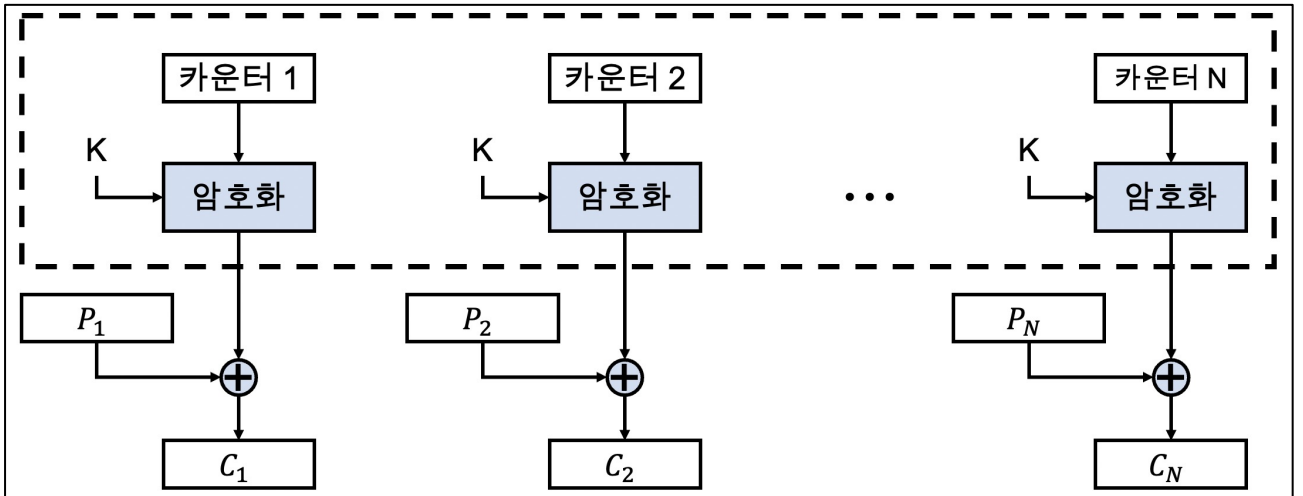
암호 블록 운용 모드

- 종류(4/4)
  - 카운터 모드(CTR: Counter Mode)
    - 정의
      - 카운터를 암호화 하고 평문 블록과 XOR하여 암호 블록을 생성하는 방식
    - 특징
      - 카운터 값이 암호화 할 각각의 평문 블록별로 달라야 함
        - 초기값으로 사용할 카운터 값을 결정한 다음에 그 다음 블록에서 사용할 카운터 값은 이전 카운터에 1을 더하여 만듦

암호 블록 운용 모드

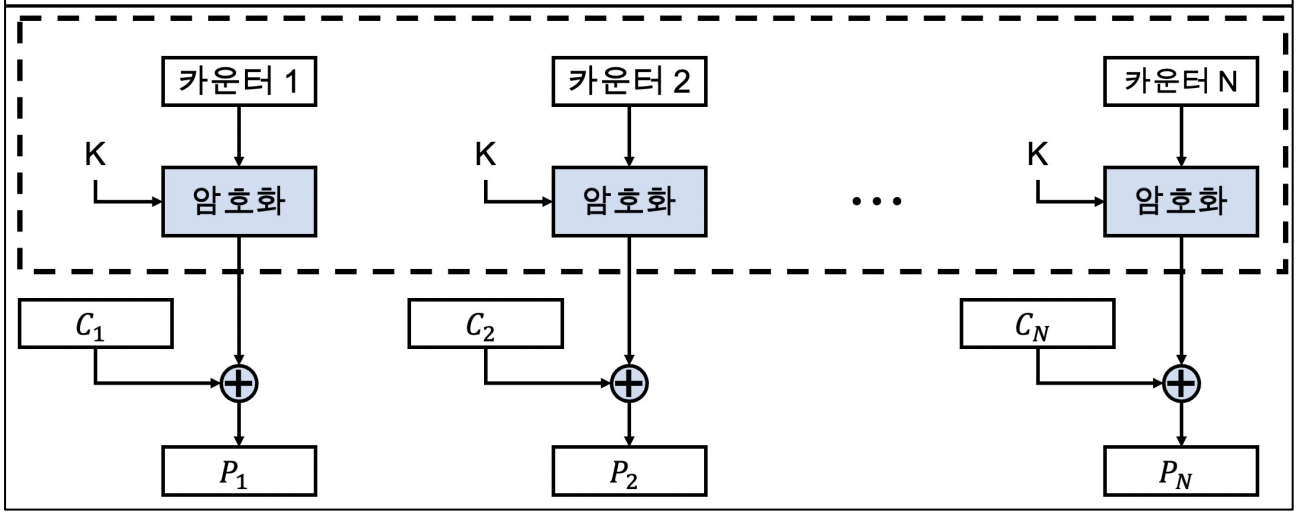
- 종류(4/4)
  - 카운터 모드(CTR: Counter Mode)
    - 암호화

$$C_i = P_i \oplus E(K, Counter)$$



    - 복호화

$$P_i = C_i \oplus E(K, Counter)$$



암호 블록 운용 모드

- 종류(4/4)
  - 카운터 모드(CTR: Counter Mode)
    - 안전성 문제
      - 한 메시지내에서 블록 단위의 패턴이 유지되지 않음
      - 암호문이 임의로 변조된다면 수신자가 복호화하는 평문에 영향을 줌
    - 응용
      - 임의 접근 가능
        - 병렬 처리 가능
      - 사전처리 가능

Thanks!

손 우 영 (wooyoung@pel.sejong.ac.kr)