

Network Security Essentials

- Chapter_2 대칭 암호와 메시지 기밀성 (2) -

발표자 : 이 태 양(taeyang@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

- 스트림 암호와 RC4
- 암호 블록 운용 모드

스트림 암호와 RC4

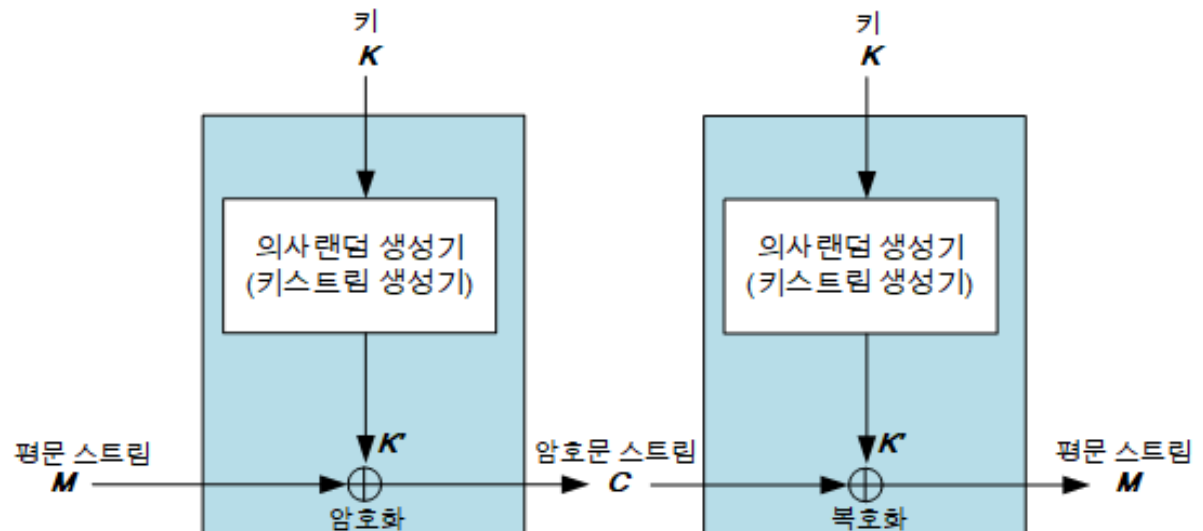
- 스트림 암호 (Stream Cipher)

- 정의

- 비트나 바이트 단위로 입력되는 요소를 연속적으로 처리하는 대칭키 암호 구조

- 구성요소

- 평문 바이트 스트림
- 사전에 공유된 키
- 암호화 알고리즘

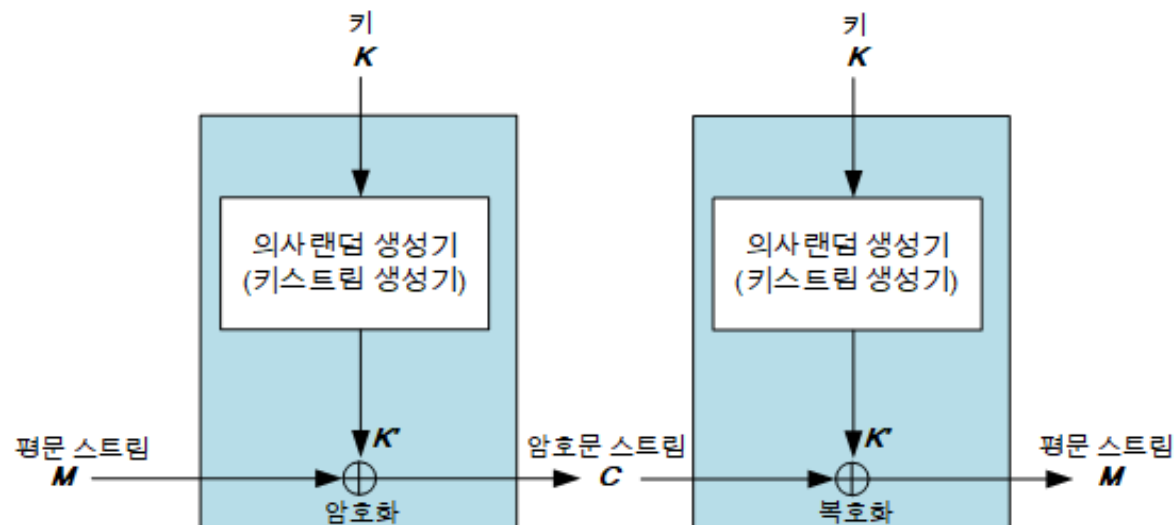


스트림 암호와 RC4

- 스트림 암호 (Stream Cipher)

- 구조

- 입력된 키로 PRNG(Pseudo Random Number Generator)의 출력값인 키스트림 생성
- 이진 평문 스트림과 이진 키스트림을 XOR 연산하여 암호문 생성
- 암호화에 사용한 키스트림과 암호문을 XOR 연산하여 복호화



스트림 암호와 RC4

- 스트림 암호 (Stream Cipher)
 - e.g, 8비트 평문 바이트 스트림과 키스트림 XOR 연산

$$\begin{array}{rcl} 11001100 & \text{평문} & \\ \oplus 01101100 & \text{키스트림} & \\ \hline 10100000 & \text{암호문} & \end{array}$$

<암호화>

$$\begin{array}{rcl} 10100000 & \text{암호문} & \\ \oplus 01101100 & \text{키스트림} & \\ \hline 11001100 & \text{평문} & \end{array}$$

<복호화>

스트림 암호와 RC4

- 스트림 암호 (Stream Cipher)
 - 스트림 암호 설계 시 고려사항
 - 암호열의 주기는 길어야 함
 - 키스트림은 진성 랜덤 스트림 특성과 근사해야 함
 - 키의 길이가 충분히 길어야 함
 - 최소 128 비트

스트림 암호와 RC4

- 스트림 암호 (Stream Cipher)

- 장점

- 이동통신 환경에서 구현이 용이
- 블록 암호보다 속도가 빠름
- 블록 암호보다 오류 발생의 영향을 덜 받음

- 단점

- 동일한 키를 가지고 두 개 이상의 평문을 암호화할 경우 해독될 가능성이 있음
- 키스트림 노출 시 해독될 가능성이 있음

스트림 암호와 RC4

- 스트림 암호 (Stream Cipher)
 - RC4 알고리즘
 - 정의
 - 내부 구현이 간단하고 빠르게 암호화 할 수 있는 초경량 암호화 알고리즘
 - 특징
 - 바이트 단위로 동작
 - 사용되는 알고리즘은 랜덤치환에 기반해서 만들어짐
 - 하나의 바이트를 출력하기 위해 8~16번의 연산이 필요
 - SSL/TLS 표준에 사용
 - WEP 프로토콜과 WPA 프로토콜에서 사용

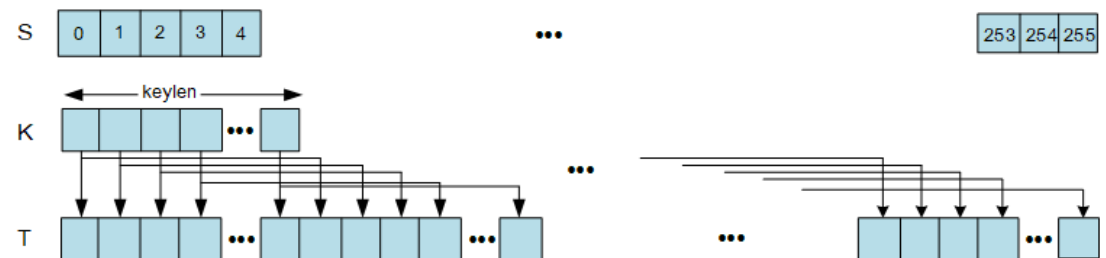
스트림 암호와 RC4

- 스트림 암호 (Stream Cipher)

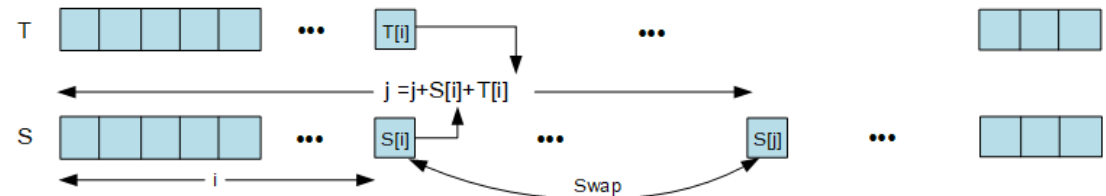
- RC4 알고리즘

- 동작 과정

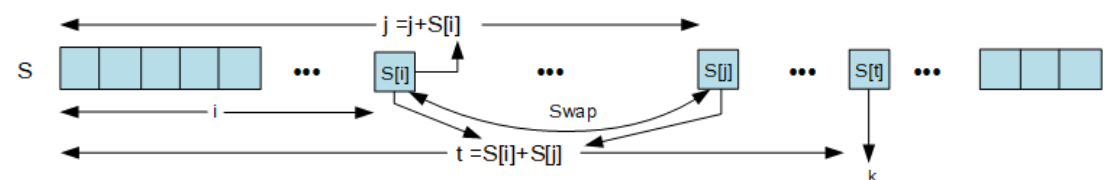
1. 벡터 S와 T의 초기화
2. 벡터 S의 초기 치환
3. 스트림 생성



1. 벡터 S와 T의 초기 상태



2. 벡터 S의 초기 치환

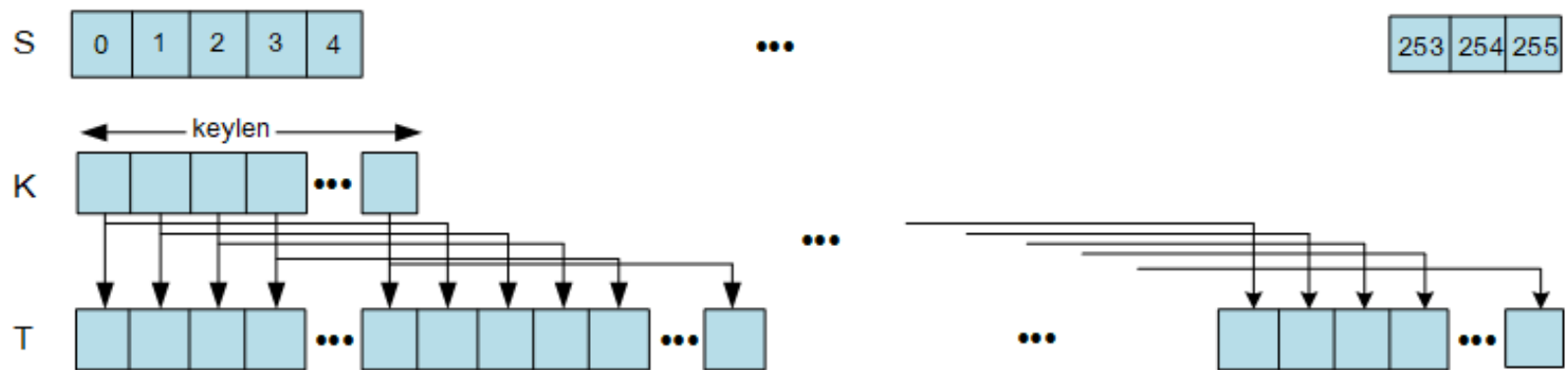


3. 스트림 생성

스트림 암호와 RC4

- 스트림 암호(Stream Cipher)
 - RC4 알고리즘
 - 동작 과정
 - 벡터 S와 T의 초기화
 - 벡터 S 각 인덱스에 0~255까지 오름차순으로 정렬
 - 벡터 T가 채워질 때까지 반복하며 키 길이만큼 벡터 T에 저장

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



1. 벡터 S와 T의 초기 상태

스트림 암호와 RC4

- 스트림 암호(Stream Cipher)

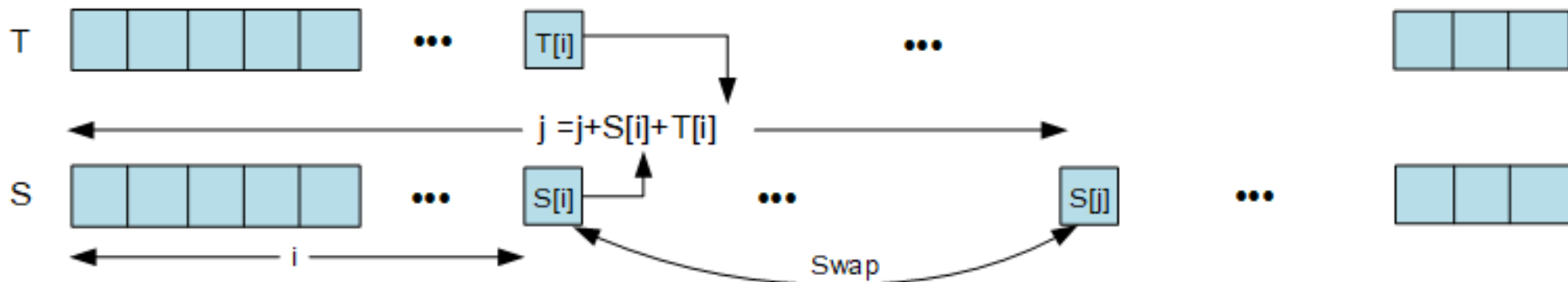
- RC4 알고리즘

- 동작 과정

- 벡터 S의 초기 치환

- S[i]와 S[j]의 위치 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]);
```



2. 벡터 S의 초기 치환

스트림 암호와 RC4

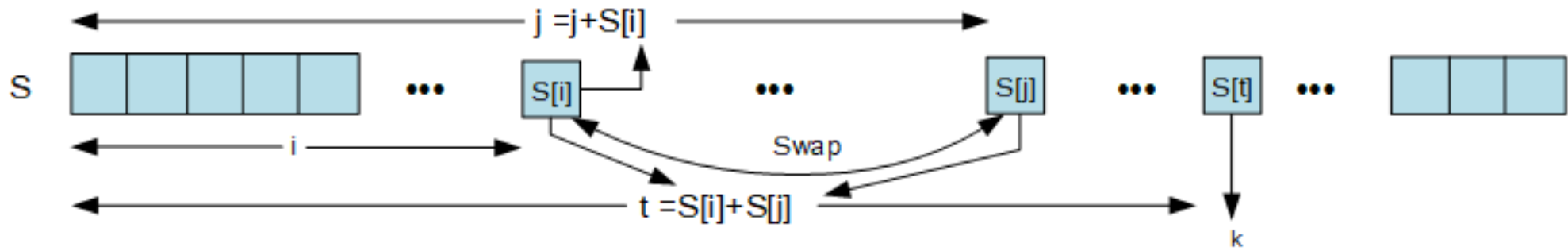
- 스트림 암호(Stream Cipher)

- RC4 알고리즘

- 동작 과정

- 키스트림 생성
 - i 는 0~255까지 값을 증가시키며 j 값을 계산
 - $S[i]$ 와 $S[j]$ 의 위치 교환
 - t 값 계산 후 $S[t]$ 값을 채우며 키스트림 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];
```



3. 스트림 생성

암호 블록 운용 모드

- 정의

- 블록암호를 반복적으로 이용하게 하는 절차
- 가변 길이 데이터를 블록 단위로 나누고 암호화하는 방법을 정하는 암호화 방식

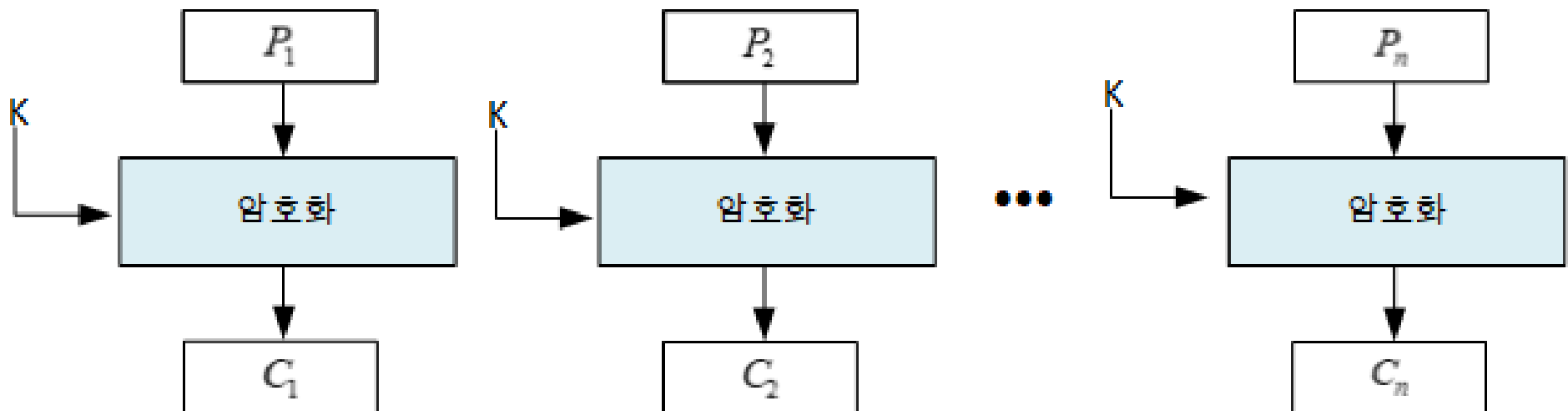
- 종류

- 전자 코드북 (ECB, Electronic CodeBook) 모드
- 암호 블록 체인 (CBC, Cipher Block Chaining) 모드
- 암호 피드백 (CFB, Cipher FeedBack) 모드
- 출력 피드백 (OFB, Output FeedBack) 모드
- 카운터 모드 (CTR, Counter) 모드

암호 블록 운용 모드

- 종류

- 전자 코드북 (ECB, Electronic CodeBook) 모드
 - 평문을 일정한 크기의 블록으로 나누고 각 블록을 동일한 키로 암호화하는 방식



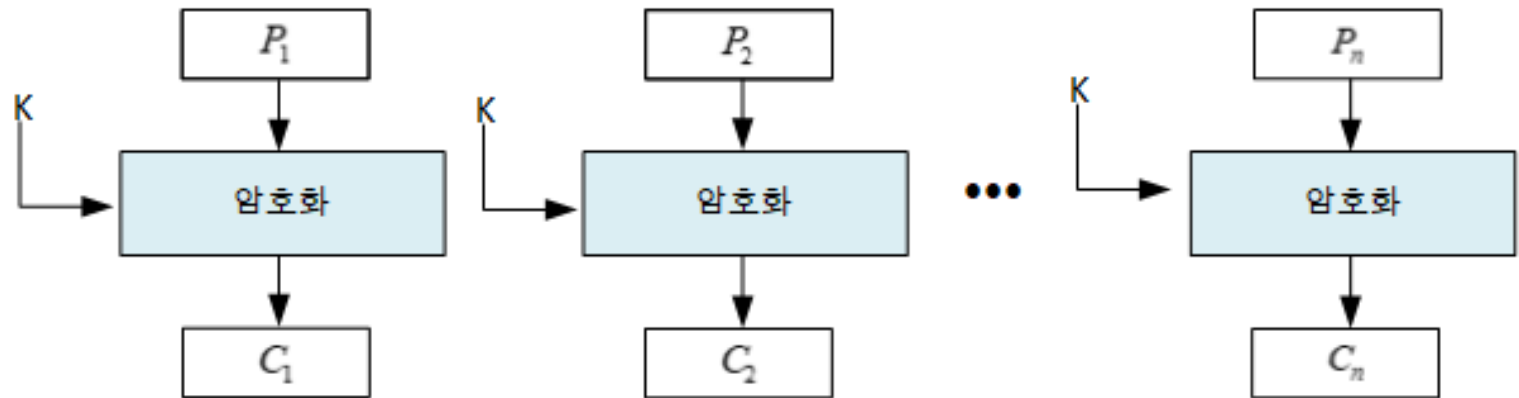
암호 블록 운용 모드

- 종류

- 전자 코드북 (ECB, Electronic CodeBook) 모드

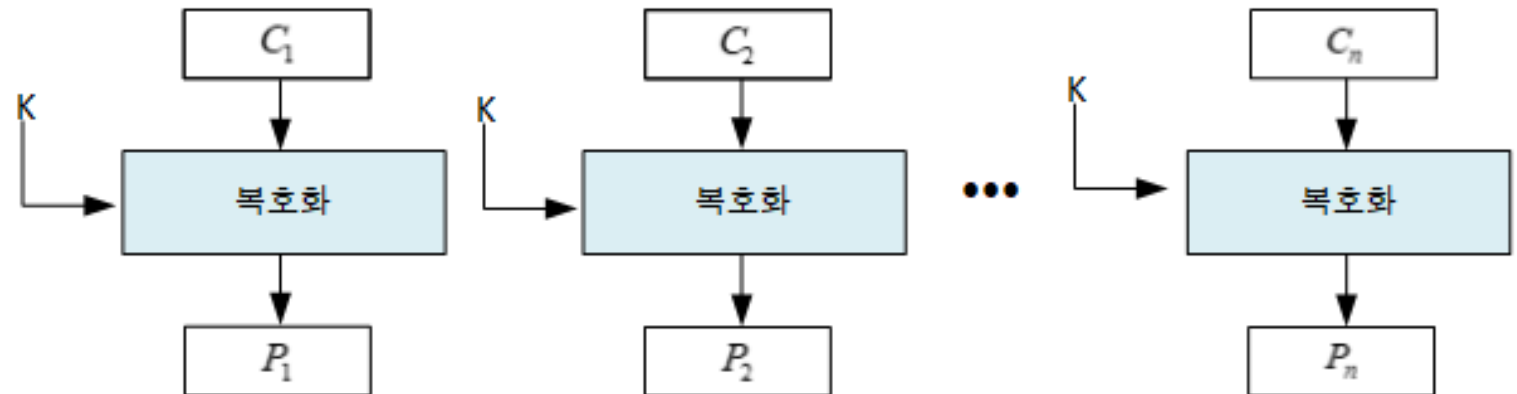
- 암호화

$$C_j = E(P_j, K)$$



- 복호화

$$P_j = D(C_j, K)$$



암호 블록 운용 모드

- 종류

- 전자 코드북 (ECB, Electronic CodeBook) 모드

- 장점

- 병렬 처리가 가능
 - 오류 확산이 없음

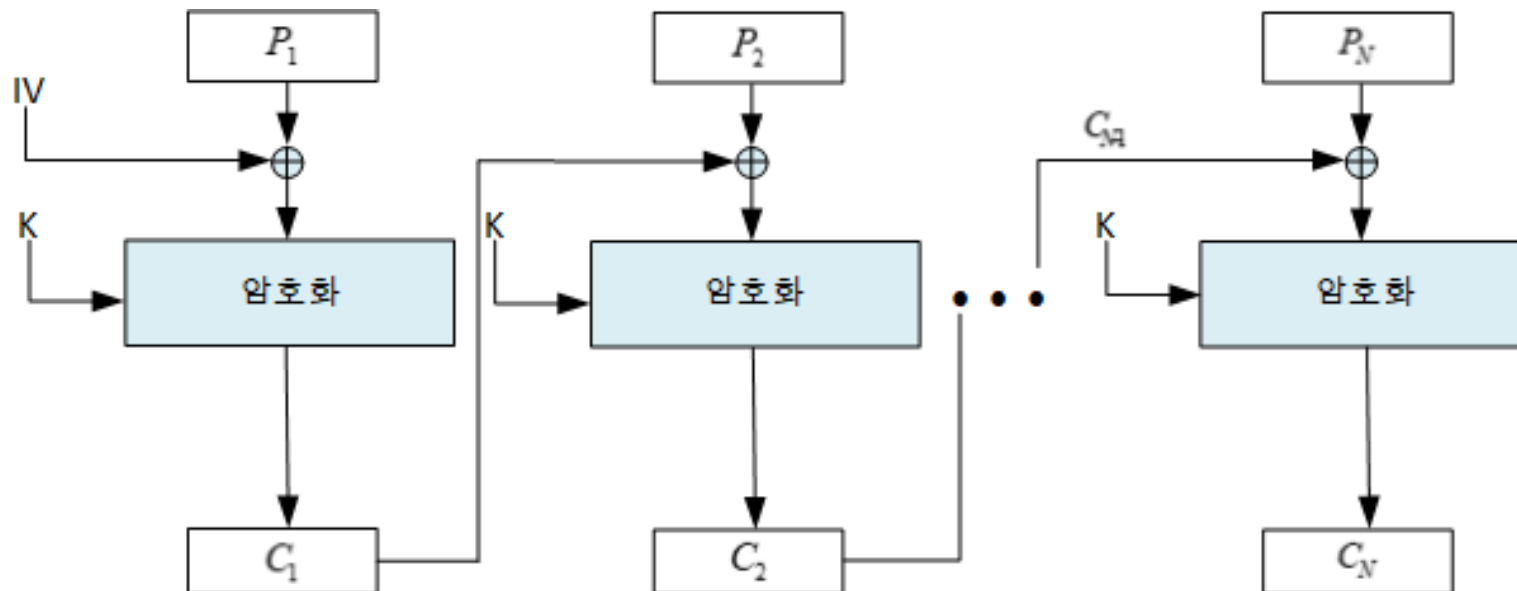
- 단점

- 패딩이 필요함
 - 블록 단위의 패턴이 드러남
 - e.g., 같은 색상의 픽셀 패턴이 드러남

암호 블록 운용 모드

- 종류

- 암호 블록 체인 (CBC, Cipher Block Chaining) 모드
 - 각 블록이 이전 블록 암호화 결과의 영향을 받으며 체인 구조를 이루는 방식
 - 첫 번째 블록 암호화는 IV(Initial Vector) 사용
 - e.g., `Aes.GenerateIV();`

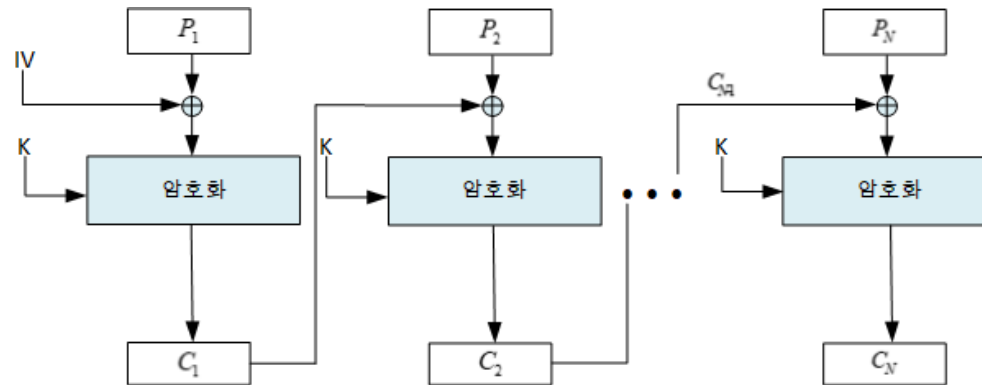


암호 블록 운용 모드

- 종류

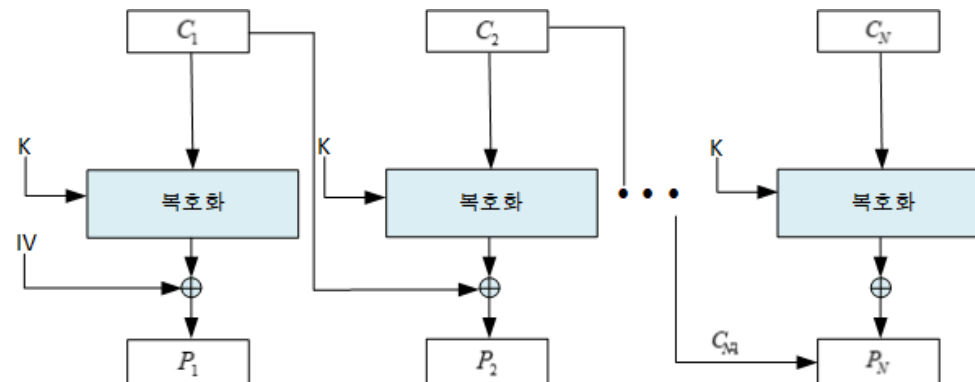
- 암호 블록 체인 (CBC, Cipher Block Chaining) 모드
 - 암호화

$$C_j = E(K[C_{j-1} \oplus P_j])$$



- 복호화

$$P_j = D(K, C_j)$$



암호 블록 운용 모드

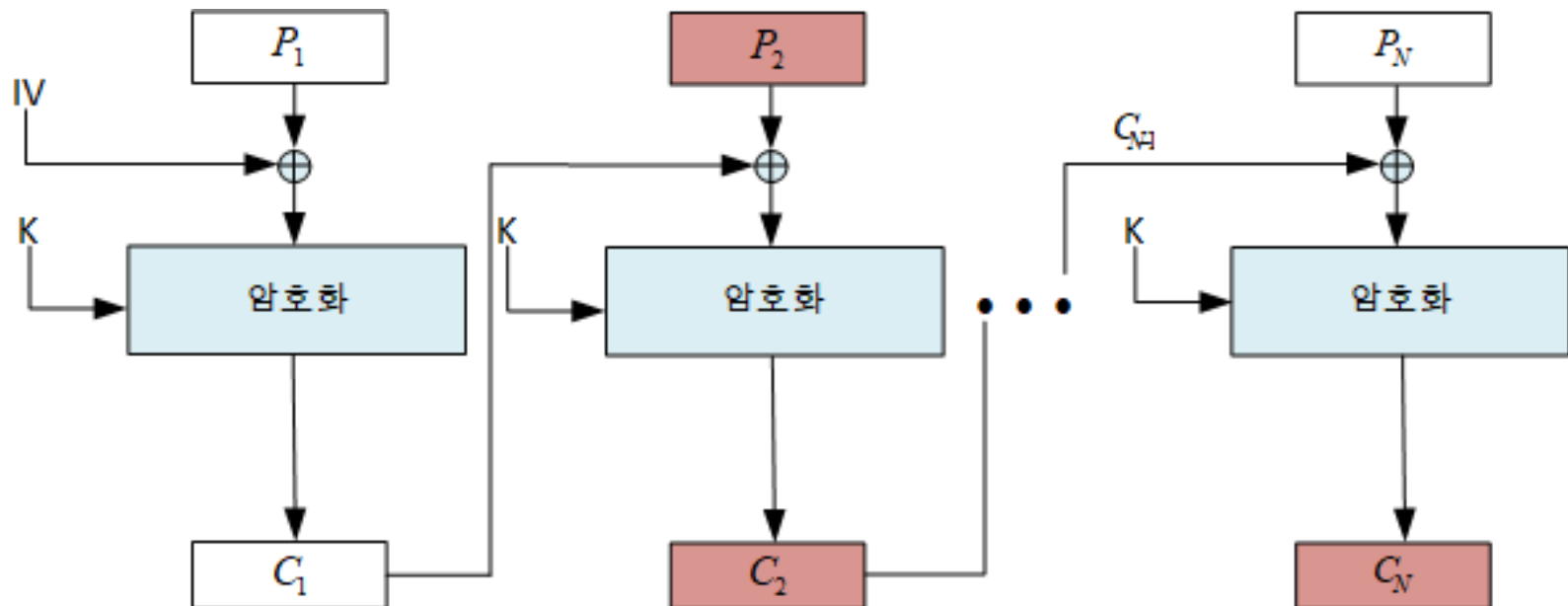
- 종류

- 암호 블록 체인 (CBC, Cipher Block Chaining) 모드

- 오류 확산

- 암호화 과정

- 평문 블록 하나에 오류가 있을 때, 현재 암호문 블록 이후 전체 암호문 블록에 영향



암호 블록 운용 모드

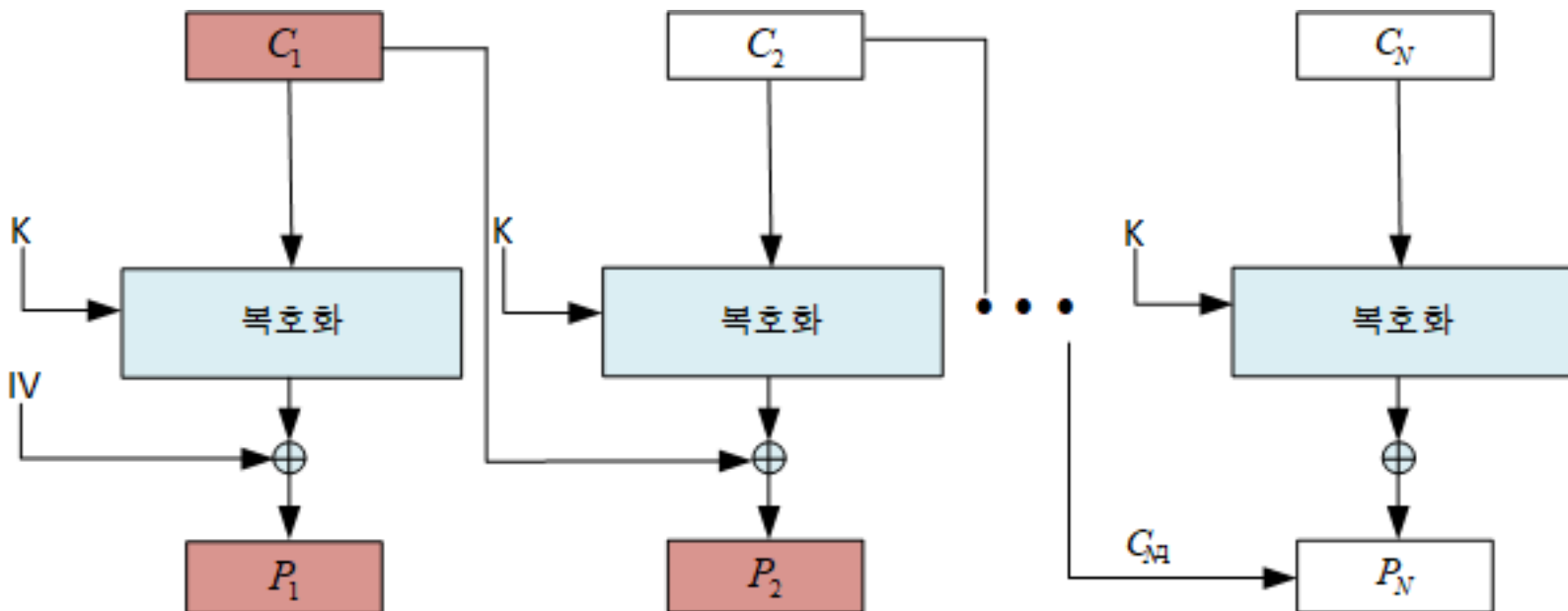
- 종류

- 암호 블록 체인 (CBC, Cipher Block Chaining) 모드

- 오류 확산

- 복호화 과정

- 암호문 블록 하나에 오류가 있을 때, 현재 평문 블록과 다음 평문 블록에만 영향



암호 블록 운용 모드

- 종류

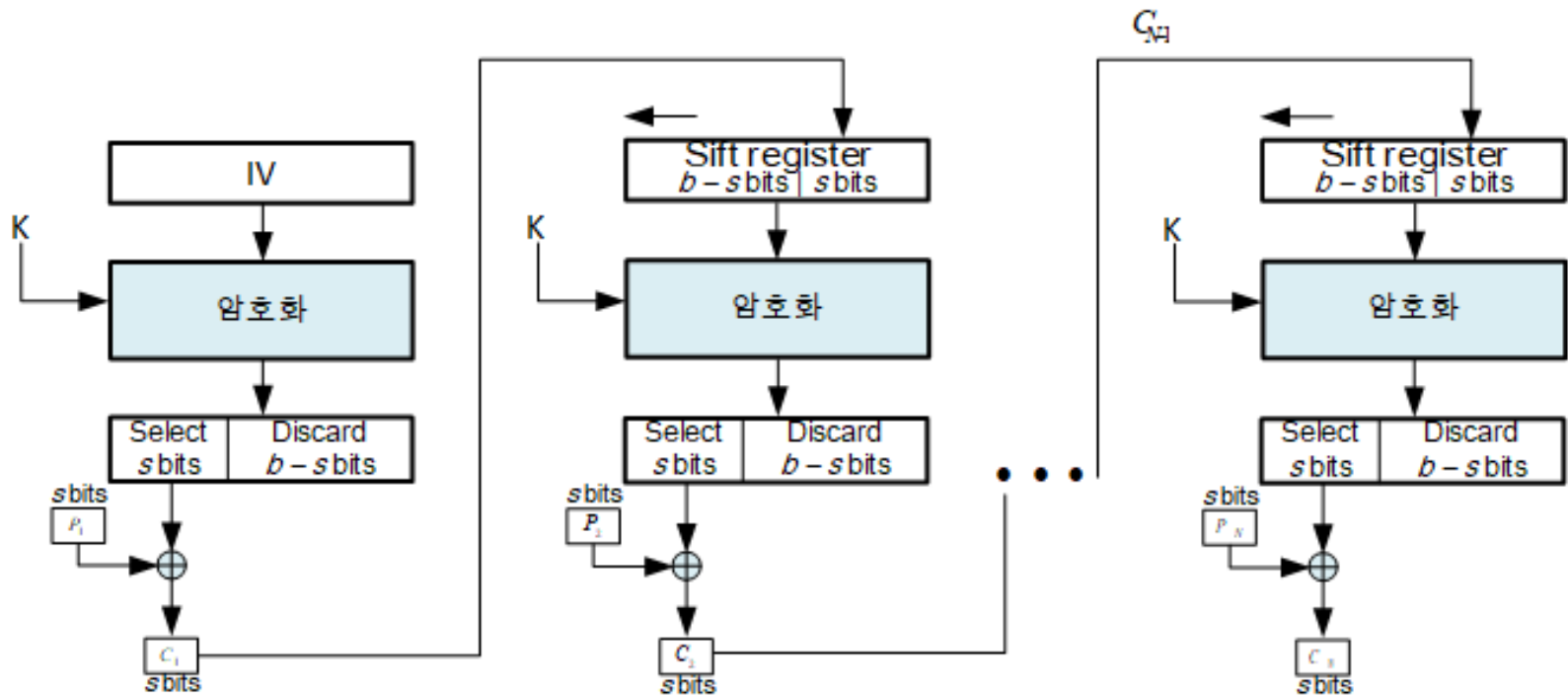
- 암호 블록 체인 모드 (CBC, Cipher Block Chaining) 모드
 - 장점
 - 평문의 반복 패턴이 드러나지 않음
 - 단점
 - 속도가 느림
 - 암호화 과정에서 병렬처리 불가능
 - 오류가 확산됨

암호 블록 운용 모드

- 종류

- 암호 피드백 (CFB, Cipher FeedBack) 모드

- 평문을 직접 암호화하지 않고 이전 암호문을 피드백하여 평문과 XOR하는 방식
- CBC 모드와 동작 방식이 비슷함



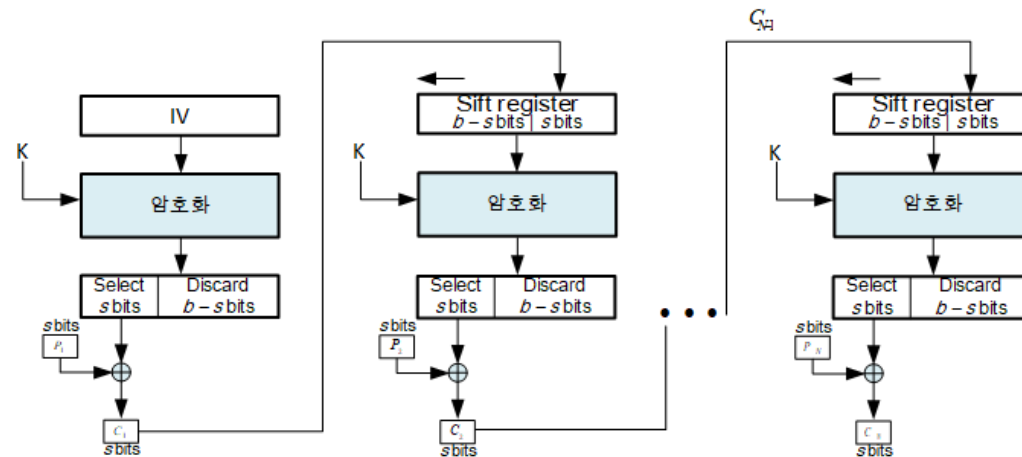
암호 블록 운용 모드

- 종류

- 암호 피드백 (CFB, Cipher FeedBack) 모드

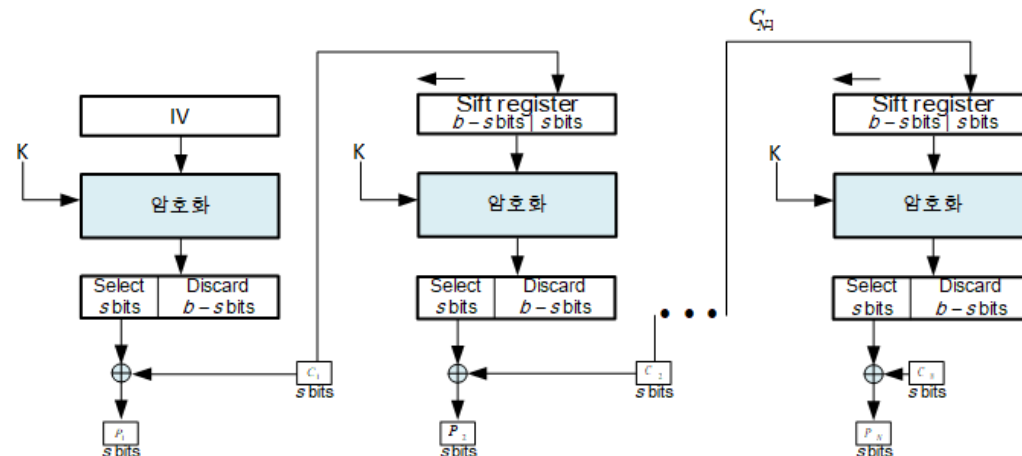
- 암호화

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



- 복호화

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



암호 블록 운용 모드

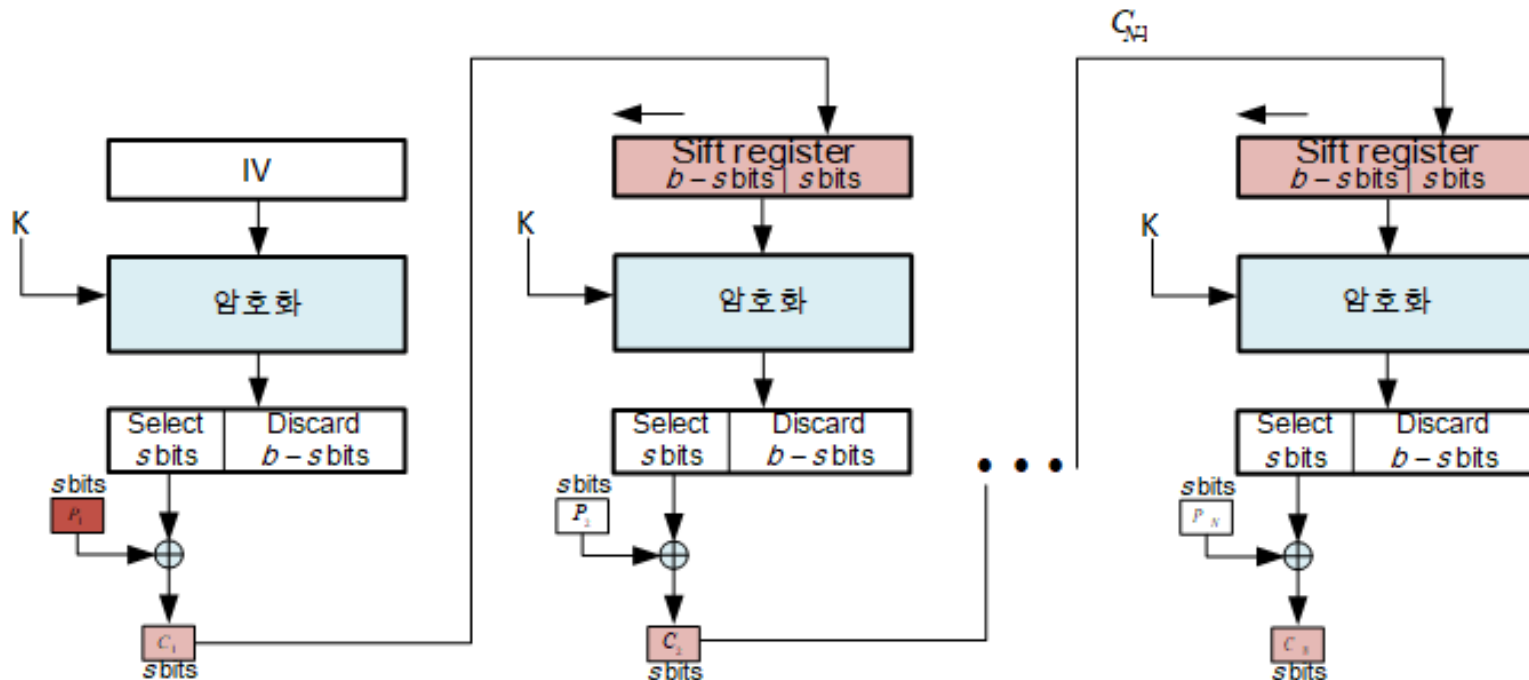
- 종류

- 암호 피드백 (CFB, Cipher FeedBack) 모드

- 오류 확산

- 암호화 과정

- 평문 블록 하나에 오류가 있다면, 현재 암호문 블록 이후 Shift register에서 오류가 소멸될 때까지 암호문에 영향



암호 블록 운용 모드

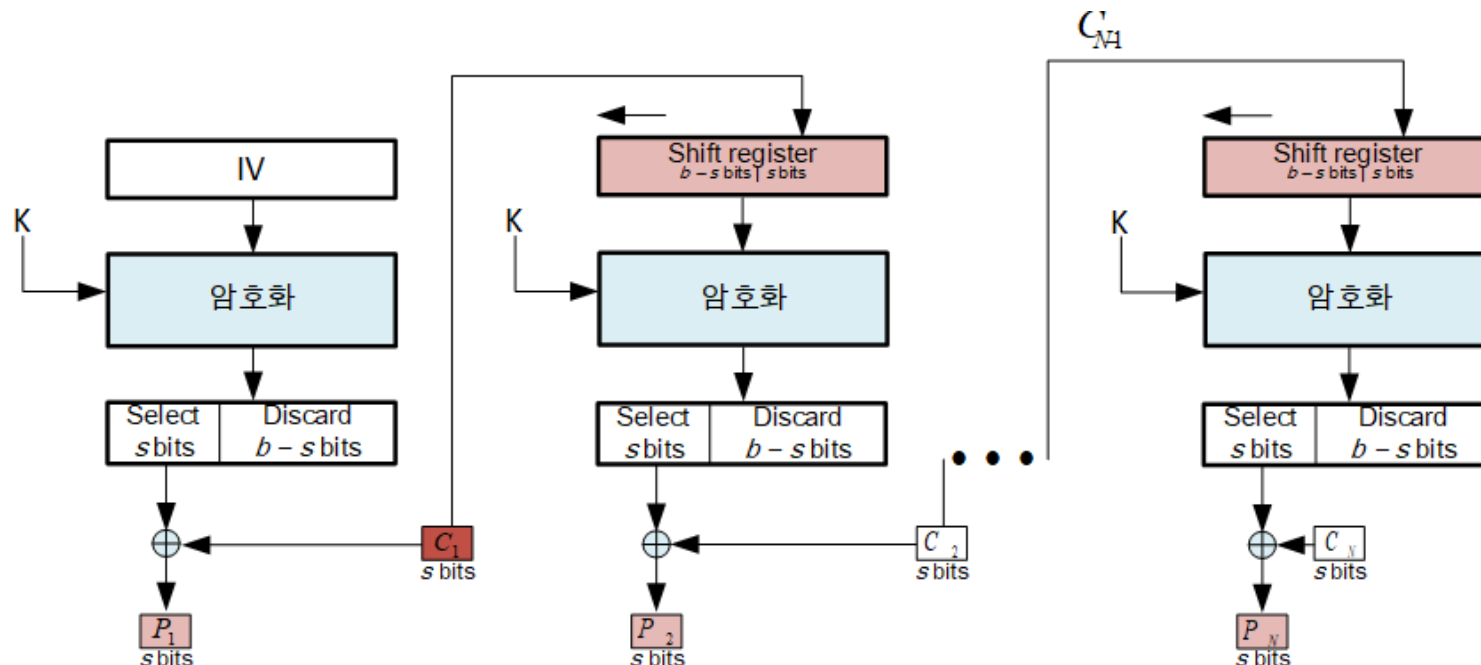
- 종류

- 암호 피드백 (CFB, Cipher FeedBack) 모드

- 오류 확산

- 복호화 과정

- 암호문 블록 하나에 오류가 있을 때, 현재 평문 블록 이후 Shift register에서 오류가 소멸 될 때까지 평문 블록에 영향



암호 블록 운용 모드

- 종류

- 암호 피드백 (CFB, Cipher FeedBack) 모드

- 장점

- 패딩이 불필요
 - 실시간 사용 가능

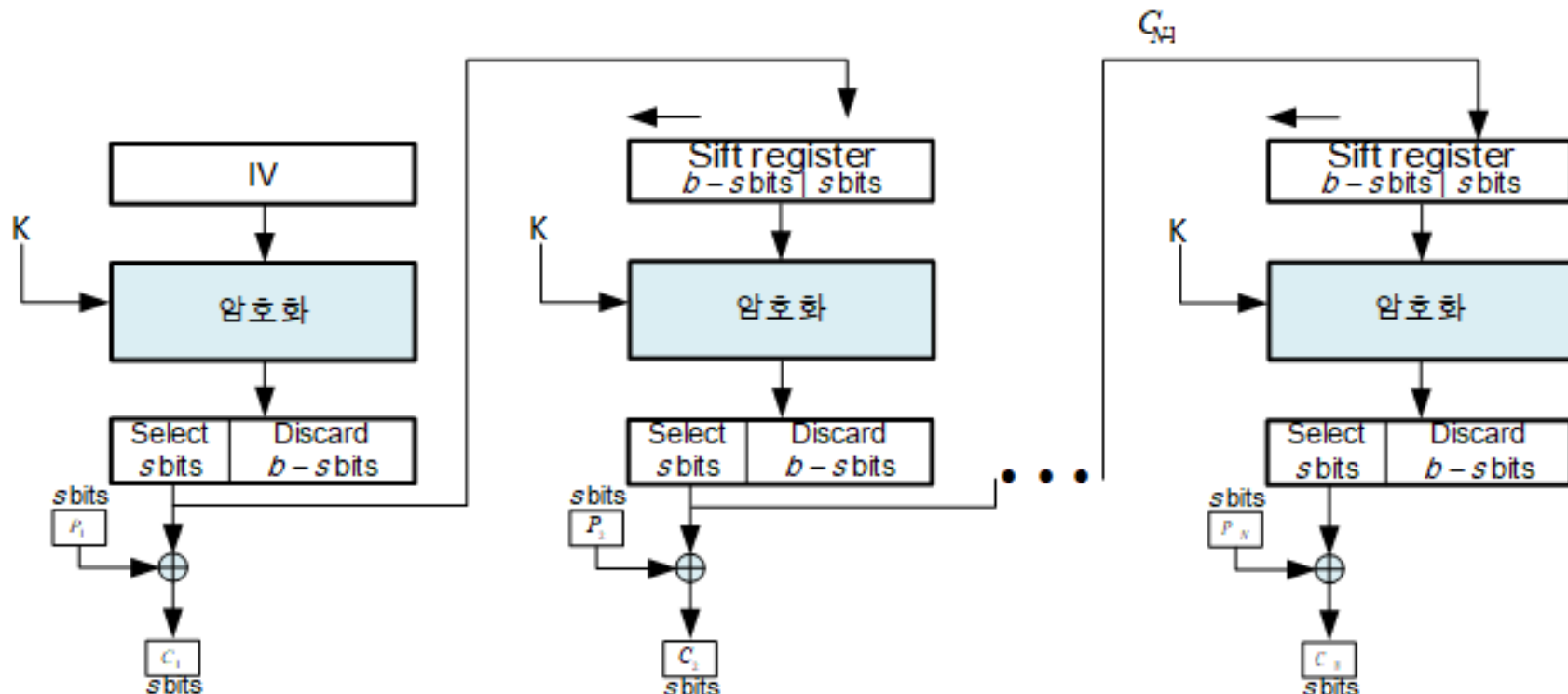
- 단점

- 속도가 느림
 - 암호화 과정에서 병렬처리 불가능
 - 오류가 확산됨

암호 블록 운용 모드

- 종류

- 출력 피드백 (OFB, Output FeedBack) 모드
 - 평문 블록에 직접 암호화하지 않고 이전 암호 알고리즘의 출력을 입력으로 피드백하여 평문과 XOR하는 방식



암호 블록 운용 모드

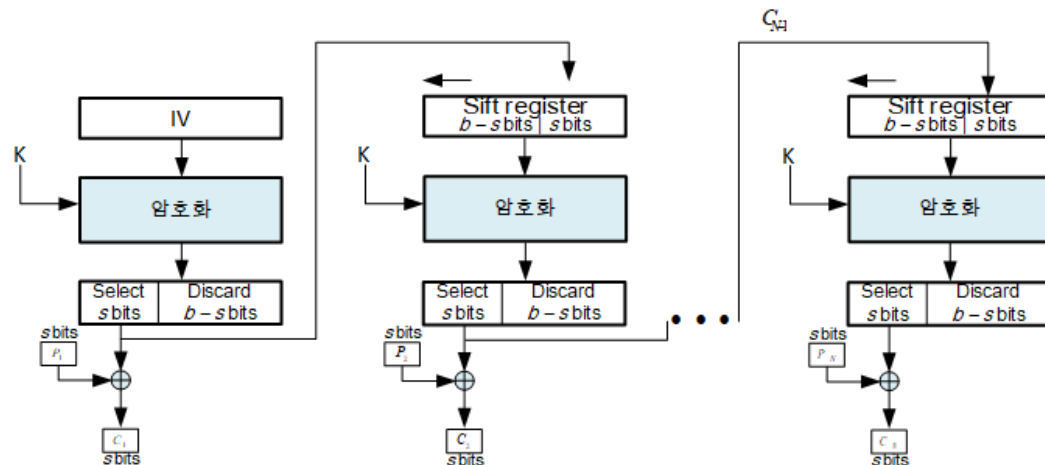
- 종류

- 출력 피드백 (OFB, Output FeedBack) 모드

- 암호화

$$P_i \oplus S_i = C_i$$

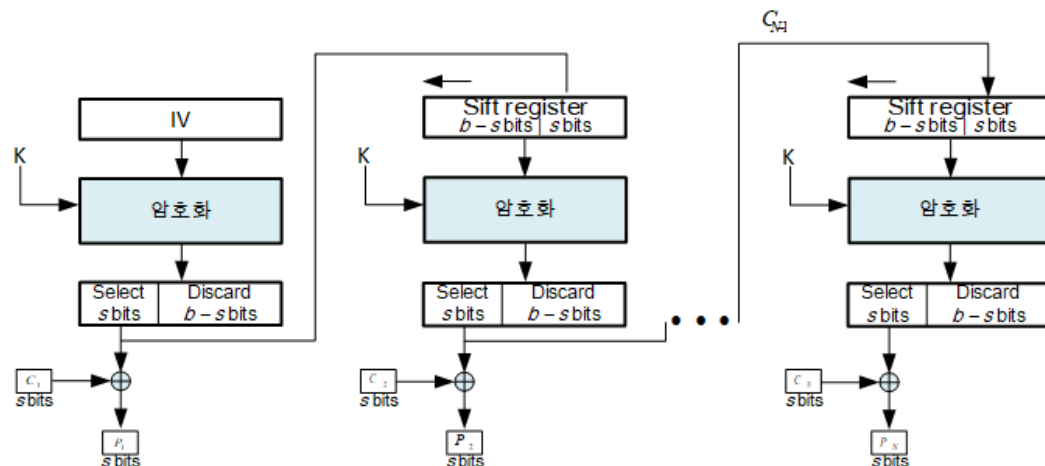
$$S_i = E_k(S_{i-1})$$



- 복호화

$$C_i \oplus S_i = P_i$$

$$S_i = E_k(S_{i-1})$$



암호 블록 운용 모드

- 종류

- 출력 피드백 (OFB, Output FeedBack) 모드

- 장점

- 패딩이 불필요
 - 오류가 확산되지 않음
 - 암호알고리즘의 출력과 평문을 XOR하여 암호문 생성

- 단점

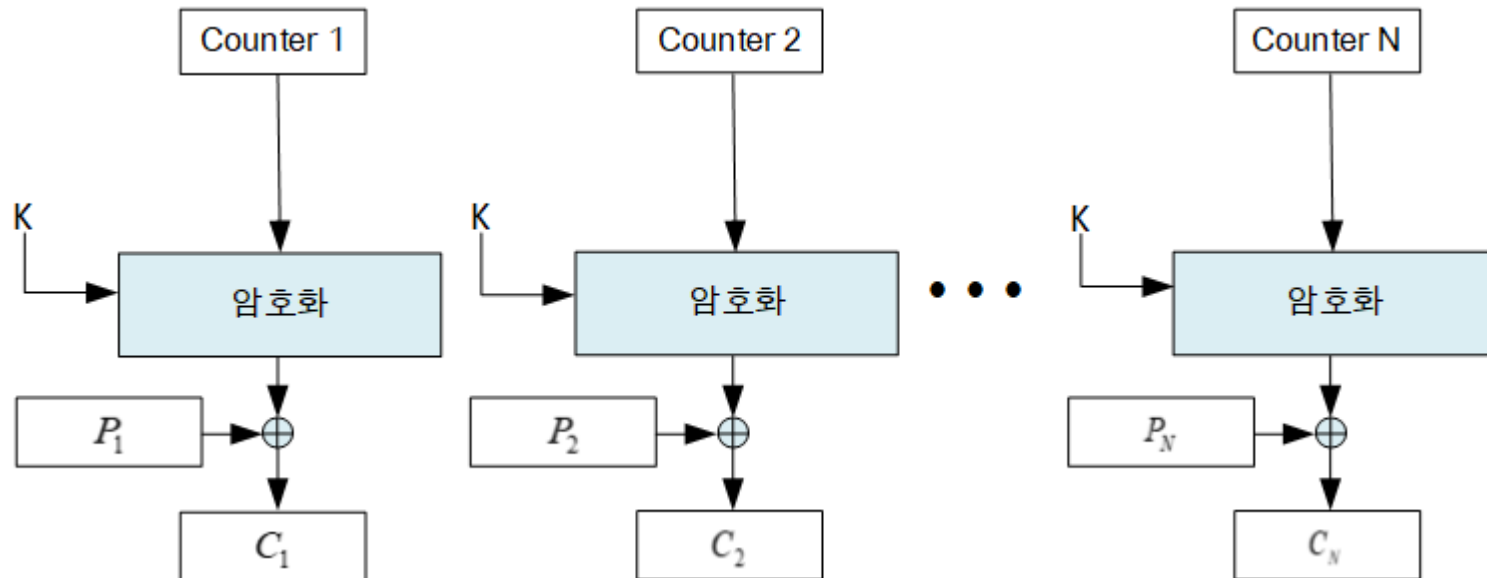
- 병렬처리 불가능

암호 블록 운용 모드

- 종류

- 카운터 (CTR, Counter) 모드

- 1씩 증가하는 카운터를 암호화해서 키스트림을 만들어내는 방식



암호 블록 운용 모드

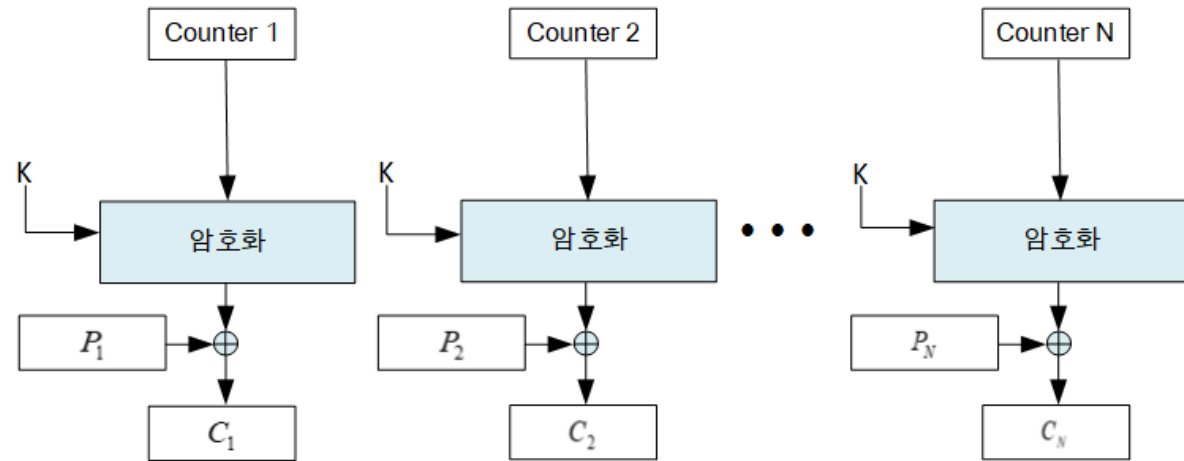
- 종류

- 카운터 (CTR, Counter) 모드

- 암호화

$$P_i \oplus S_i = C_i$$

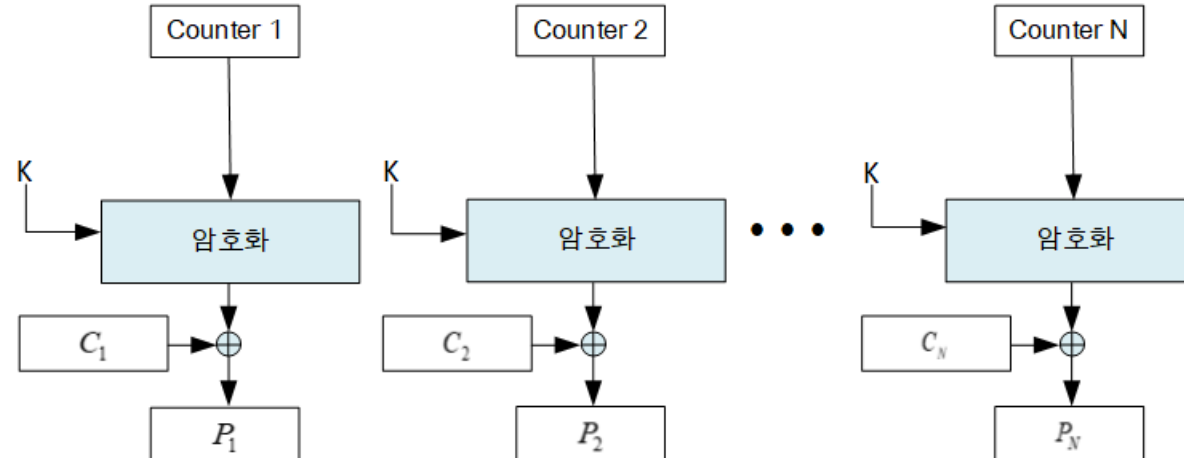
$$S_i = E_k(N_i)$$



- 복호화

$$C_i \oplus S_i = P_i$$

$$S_i = E_k(N_i)$$



암호 블록 운용 모드

- 종류

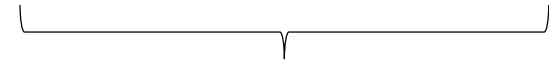
- 카운터 (CTR, Counter) 모드
 - 카운터 구조

66 1F 98 CD 37 A3 8B 4B



비표

00 00 00 00 00 00 00 01



블록 번호

평문 블록 1의 카운터 66 1F 98 CD 37 A3 8B 4B

00 00 00 00 00 00 00 01

평문 블록 2의 카운터 66 1F 98 CD 37 A3 8B 4B

00 00 00 00 00 00 00 02

평문 블록 3의 카운터 66 1F 98 CD 37 A3 8B 4B

00 00 00 00 00 00 00 03

암호 블록 운용 모드

- 종류

- 카운터 (CTR, Counter) 모드

- 장점

- 병렬처리 가능
 - 오류가 확산되지 않음
 - 사전에 카운터를 미리 암호화할 수 있음
 - 원하는 부분만 복호화할 수 있음
 - 처리속도가 빠름

- 단점

- 공격자가 암호문 블록의 비트를 반전시키면 대응하는 평문 블록 비트가 반전됨

암호 블록 운용 모드

• 암호 블록 운용 모드 비교

운용 모드	병렬처리	패딩	초기화 벡터	오류확산
ECB	○	○	X	X
CBC	복호화만	○	○	E : 해당 블록 이후 모든 블록 D : 해당 블록 다음 블록
CFB	복호화만	X	○	Shift register에서 오류가 완전히 소멸 될 때까지
OFB	X	X	○	X
CTR	○	X	X	X

Thanks!

이 태 양 (taeyang@pel.sejong.ac.kr)