

Network Security Essentials

- Chapter_2 대칭 암호와 메시지 기밀성 -

발표자 : 이 태 양(taeyang@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

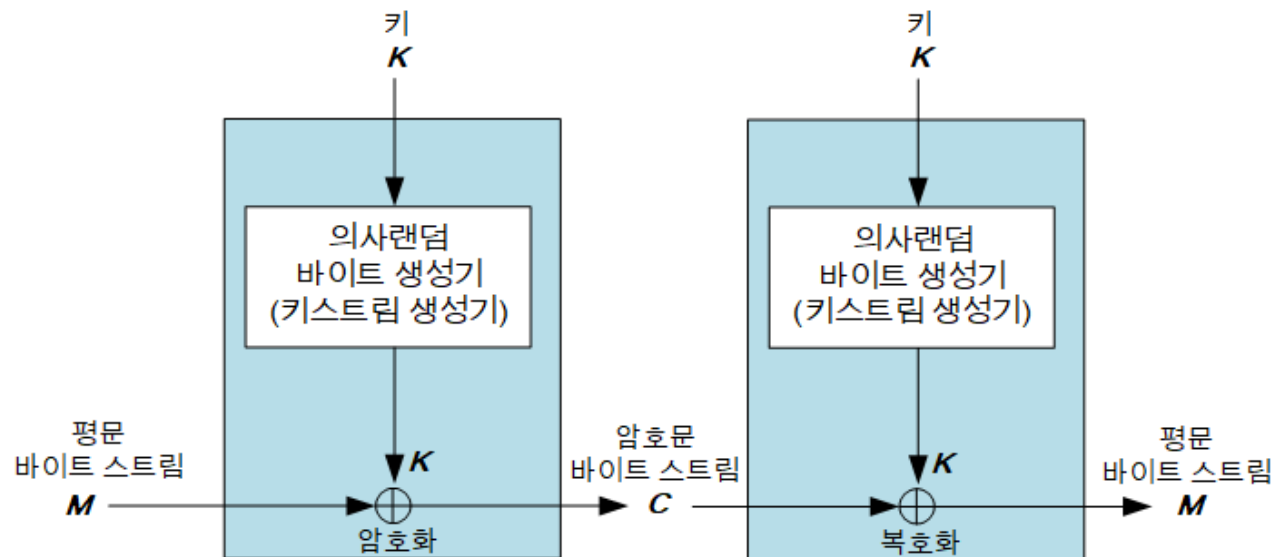
- 스트림 암호와 RC4
- 암호 블록 운용 모드

스트림 암호와 RC4

- 스트림 암호 (Stream Cipher)

- 구성요소

- 평문 바이트 스트림
- 키
- 암호화 알고리즘

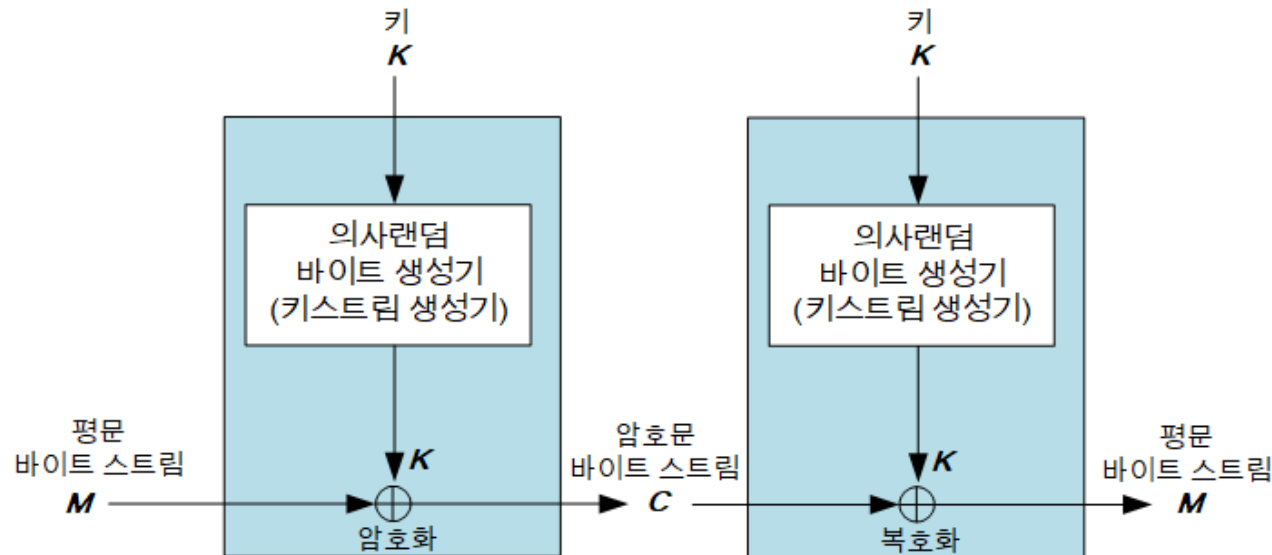


스트림 암호와 RC4

- 스트림 암호 (Stream Cipher)

- 구조

- 입력된 키로 PRNG의 출력값인 키스트림 생성
- 평문 바이트 스트림과 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 스트림과 근사해야 함
 - 키의 길이가 충분히 길어야 함
 - 최소 128 비트

스트림 암호와 RC4

- 스트림 암호 (Stream Cipher)

- 장점

- 실용적이며, 군사적인 목적으로 많이 사용
- 이동통신 환경에서 구현이 용이
- 블록 암호보다 적은 양의 코드를 사용하고 속도가 빠름
- 블록 암호보다 오류 발생의 영향을 덜 받음

- 단점

- 동일한 키를 가지고 두 개 이상의 평문을 암호화할 경우 해독이 가능할 수도 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 작동
 - 사용되는 알고리즘은 랜덤치환에 기초해서 만들어짐
 - 하나의 바이트를 출력하기 위해 8~16번의 연산이 필요
 - SSL/TLS 표준에 사용
 - WEP 프로토콜과 WPA 프로토콜에서 사용

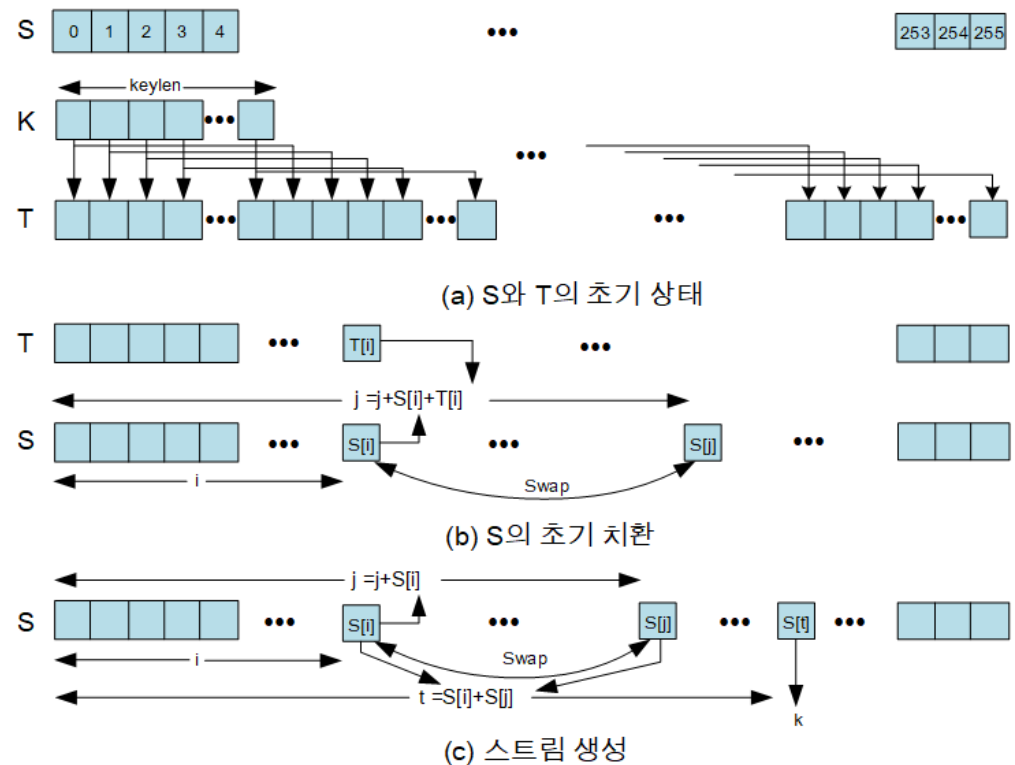
스트림 암호와 RC4

- 스트림 암호 (Stream Cipher)

- RC4 알고리즘

- 동작 과정

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



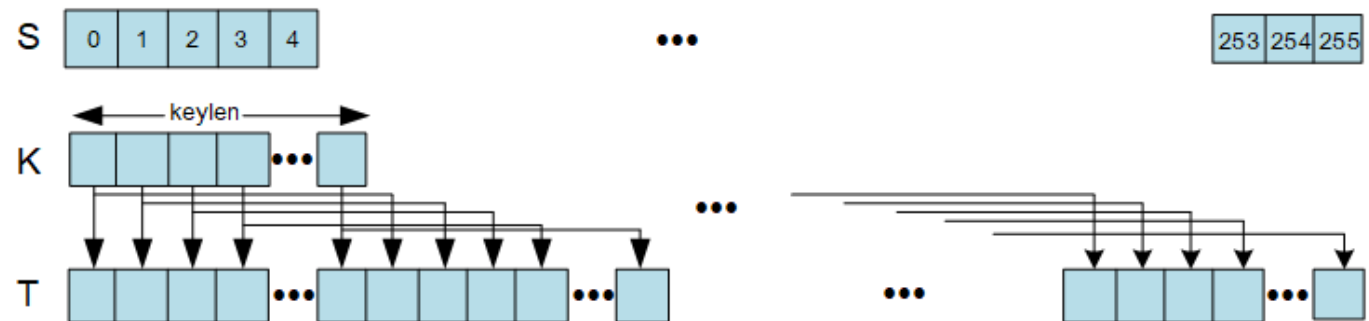
스트림 암호와 RC4

- RC4 알고리즘

- 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];
```



(a) S와 T의 초기 상태

스트림 암호와 RC4

- RC4 알고리즘
 - 벡터 S의 초기 치환
 - $S[i]$ 와 $S[j]$ 의 위치 교환

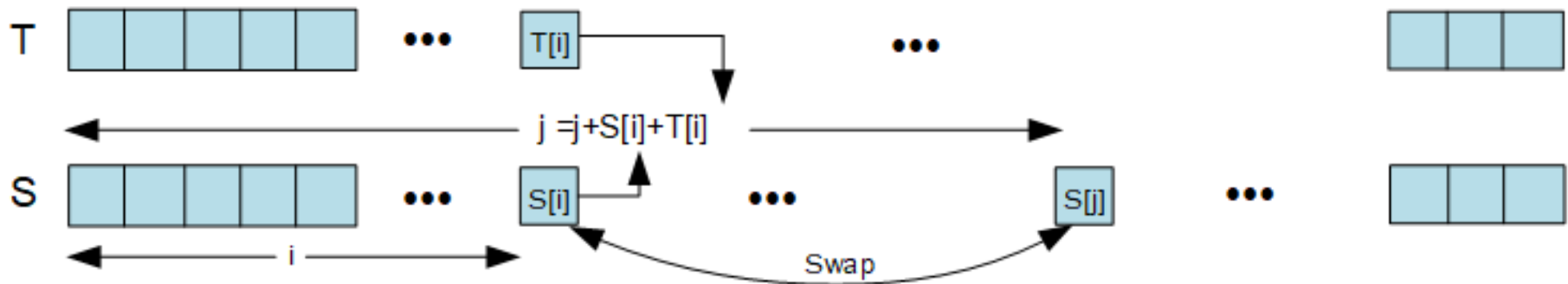
/ Initial Permutation of S */*

$j = 0;$

for $i = 0$ to 255 do

$j = (j + S[i] + T[i]) \bmod 256;$

Swap ($S[i], S[j]$);



(b) S의 초기 치환

스트림 암호와 RC4

- RC4 알고리즘

- 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)

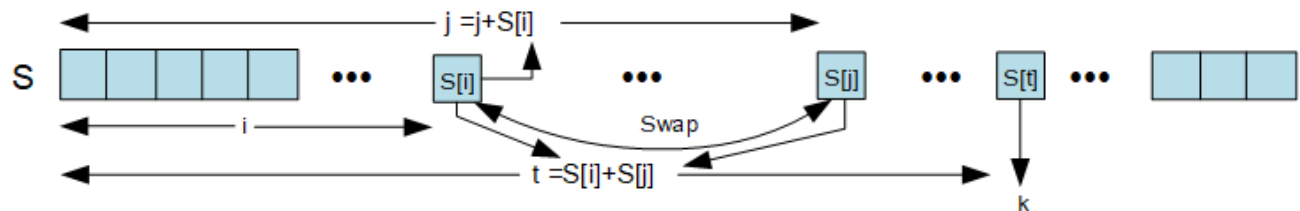
$i = (i+1) \bmod 256;$

$j = (j+S[i]) \bmod 256;$

Swap ($S[i], S[j]$);

$t = (S[i] + S[j]) \bmod 256;$

$K = S[t];$



(c) 스트림 생성

암호 블록 운용 모드

- 정의

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

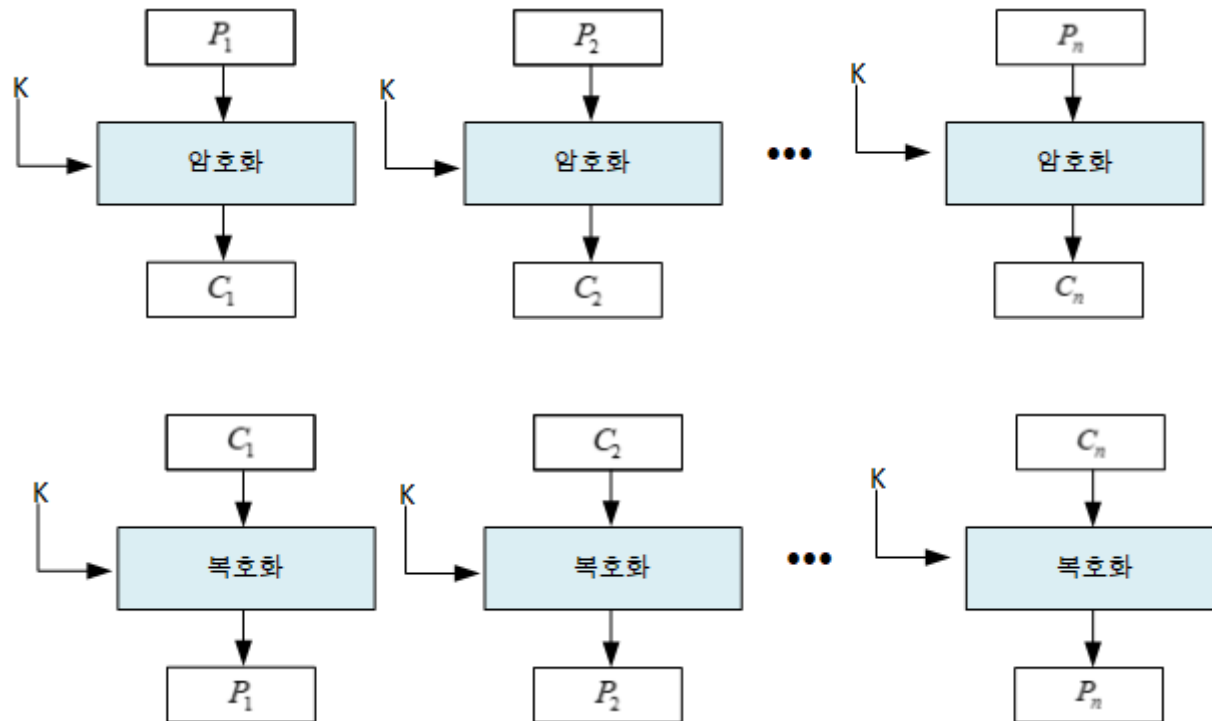
- 종류

- 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

- 장점

- 병렬 처리가 가능
 - 오류 확산이 없음

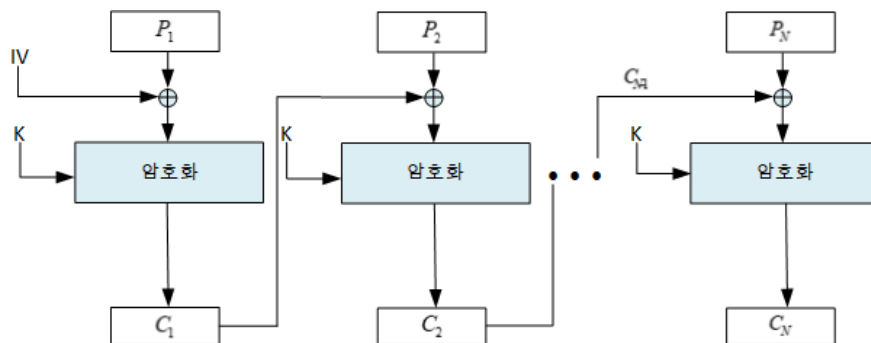
- 단점

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

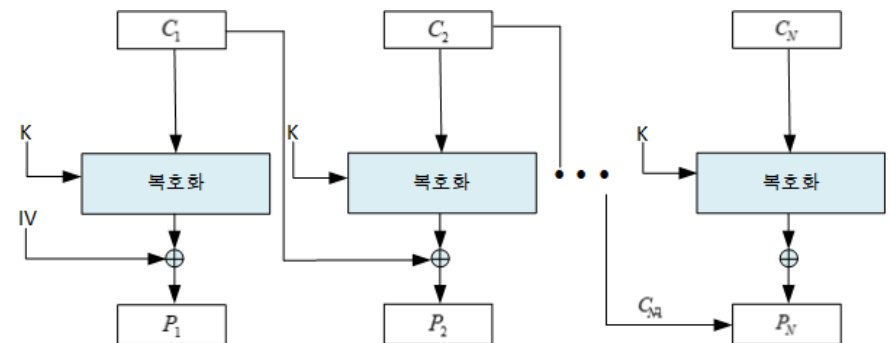
암호 블록 운용 모드

- 종류

- 암호 블록 체인 모드 (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



<암호화>



<복호화>

암호 블록 운용 모드

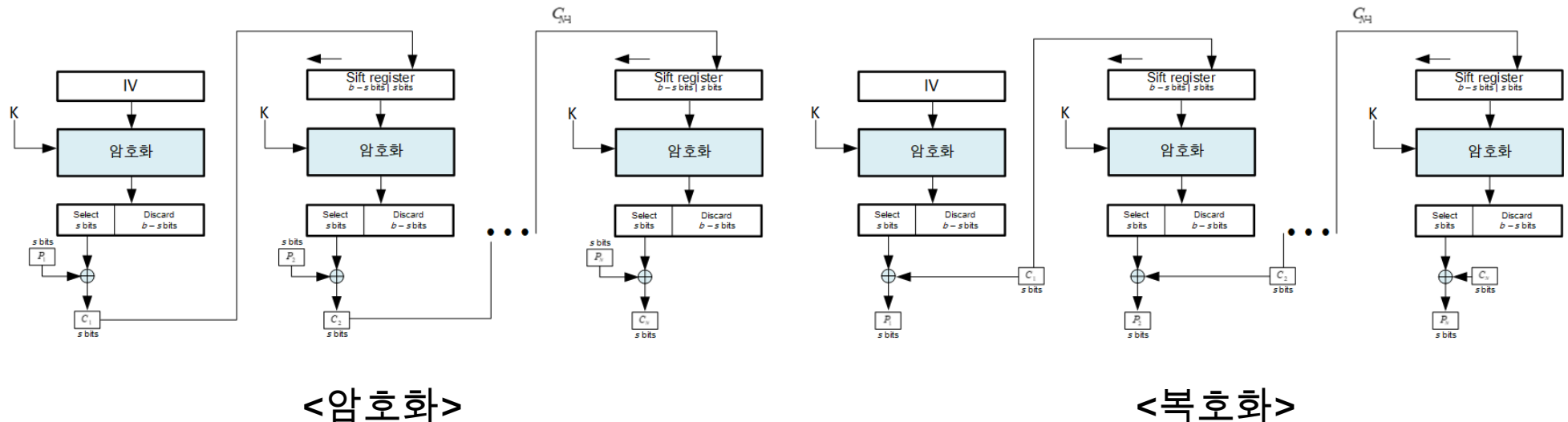
- 종류

- 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 암호화하여 평문과 XOR하는 방식
 - CBC 모드와 동작 방식이 비슷함



암호 블록 운용 모드

- 종류

- 암호 피드백 (CFB, Cipher FeedBack) 모드

- 장점

- 패딩이 불필요
 - 실시간 사용 가능

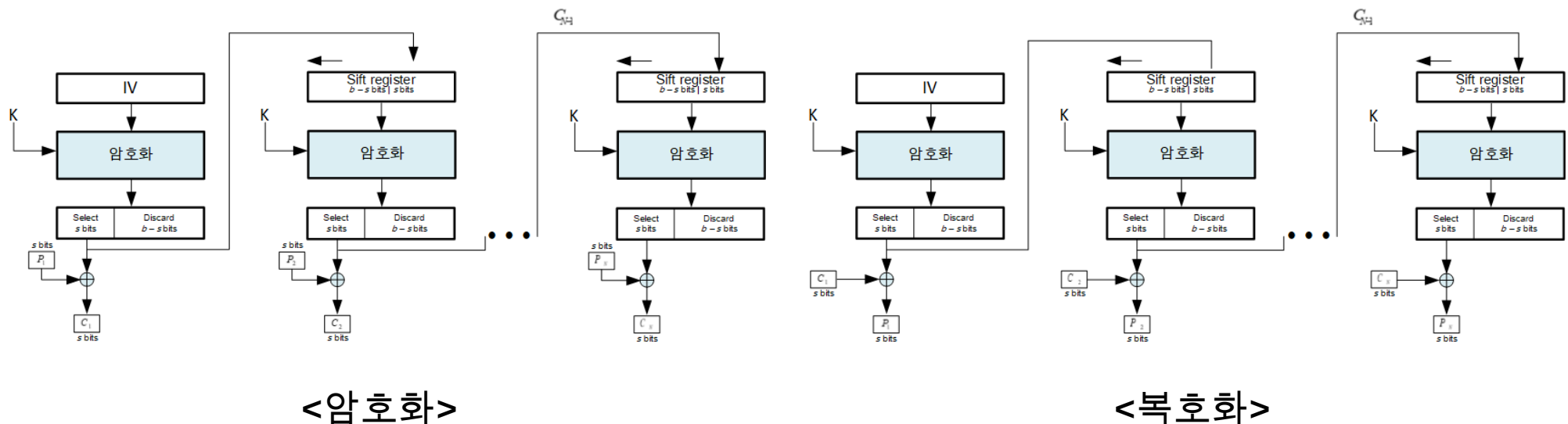
- 단점

- 속도가 느림
 - 암호화 과정에서 병렬처리 불가능
 - 오류가 확산됨
 - 암호화의 경우 오류가 있는 암호문 블록 이후 오류가 소멸될 때까지 암호문에 영향
 - 복호화의 경우 오류가 있는 평문 블록 이후 오류가 소멸될 때까지 평문 블록에 영향

암호 블록 운용 모드

- 종류

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



암호 블록 운용 모드

- 종류

- 출력 피드백 (OFB, Output FeedBack) 모드

- 장점

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

- 단점

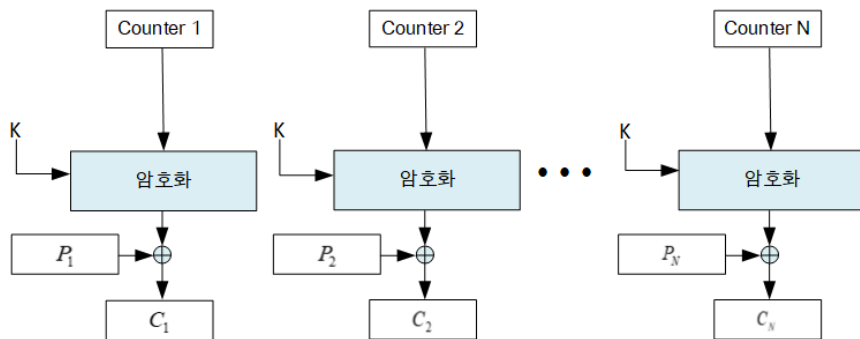
- 병렬처리 불가능

암호 블록 운용 모드

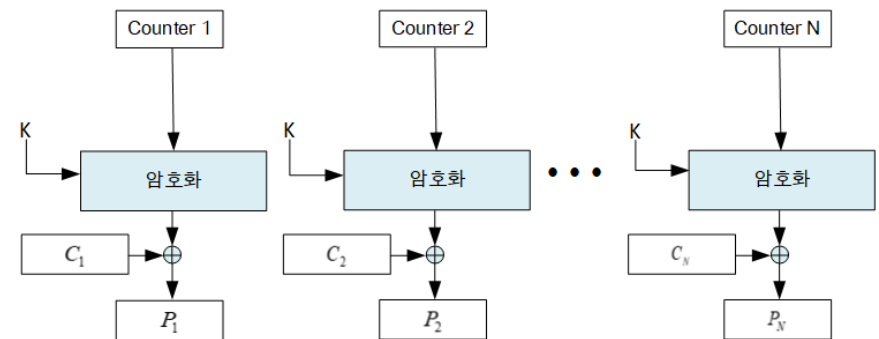
- 종류

- 카운터 (CTR, Counter) 모드

- 1씩 증가하는 카운터를 암호화해서 키스트림을 만들어내는 방식



<암호화>



<복호화>

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

Thanks!

이 태 양 (taeyang@pel.sejong.ac.kr)