제2장. 대칭 암호와 메시지 기밀성

2.4 스트림 암호와 RC4

스트림 암호

- 블록(Block) 암호
  - 블록 단위로 처리하여 각 입력 블록에 대응하는 출력 블록 생성
- 스트림(Stream) 암호
  - 입력되는 요소를 연속적으로 처리하여 지속적으로 출력

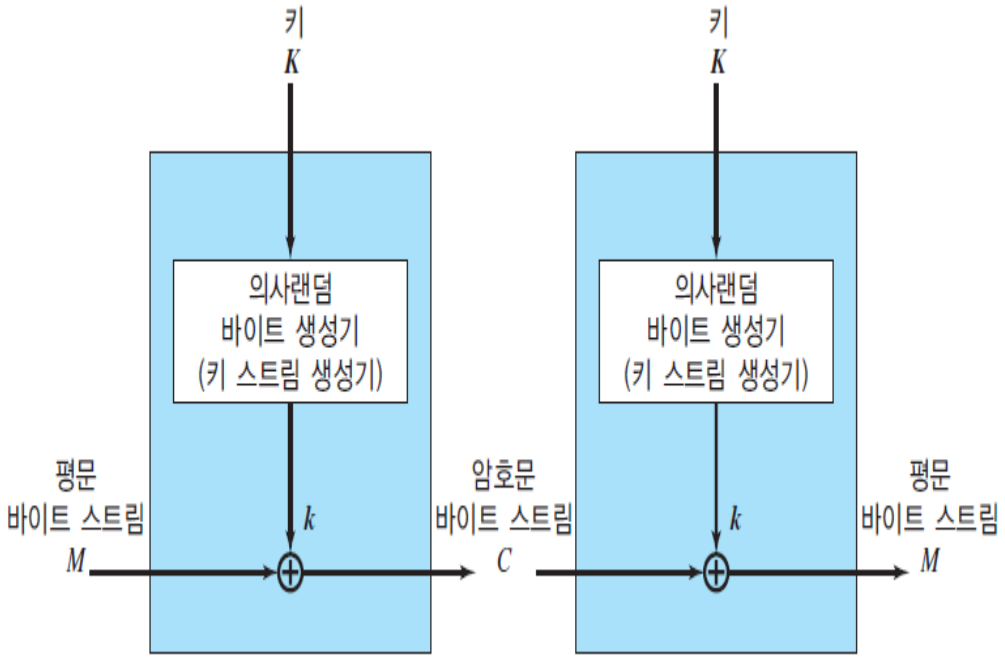
스트림 암호 구조 (1)

- 암호화

11001100	평문
⊕ 01101100	키 스트림
10100000	암호문

- 복호화

10100000	암호문
⊕ 01101100	키 스트림
11001100	평문



스트림 암호 구조 (2)

- 스트림 암호 설계시 주의사항
  - 주기가 긴 암호열
    - 의사랜덤수 생성기의 결과는 반복적으로 나타나는 결정적 비트 스트림
  - 진성랜덤넘버 특성에 근사한 키 스트림
  - 충분히 긴 키의 길이
- 적절히 설계된 의사 랜덤넘버생성기로 구현한 스트림 암호는 동등한 길이의 키를 사용하는 블록암호만큼의 보안성 유지 가능

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 구조 (3)

- 장점
  - 속도면에서 월등히 빠름
  - 작은 양의 코드로 구현
  - 데이터 통신 채널과 같이 스트림 암호화가 필요한 경우 적합
- 단점
  - 2개의 평문을 동일한 키로 암호화하는 경우 암호 해독이 아주 단순해짐
    - 두 개의 암호문 XOR = 원래의 평문과의 XOR

RC4 알고리즘 (1)

- 론 리베스트(Ron Rivest)가 1987년에 RSA Security에서 설계한 스트림 암호
- 바이트 단위로 작동하는 다양한 크기의 키 사용
- 랜덤 치환 기법 사용
- 암호 주기가 10^{100}보다 큼
- 응용
  - SSL/TLS
  - WEP(Wired Equivalent Privacy) 프로토콜
  - WPA(WiFi Protocol Access) 프로토콜
- 원래 알고리즘을 비밀로 하였으나, 1994년 익명의 제보자가 공개

RC4 알고리즘 (2)

- RC 4에서 S(상태 벡터) 초기화

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

T: 임시벡터

S[i]: S의 성분 i= 0부터 255

RC4 알고리즘 (3)

- RC 4에서 T를 사용한 S 초기치환

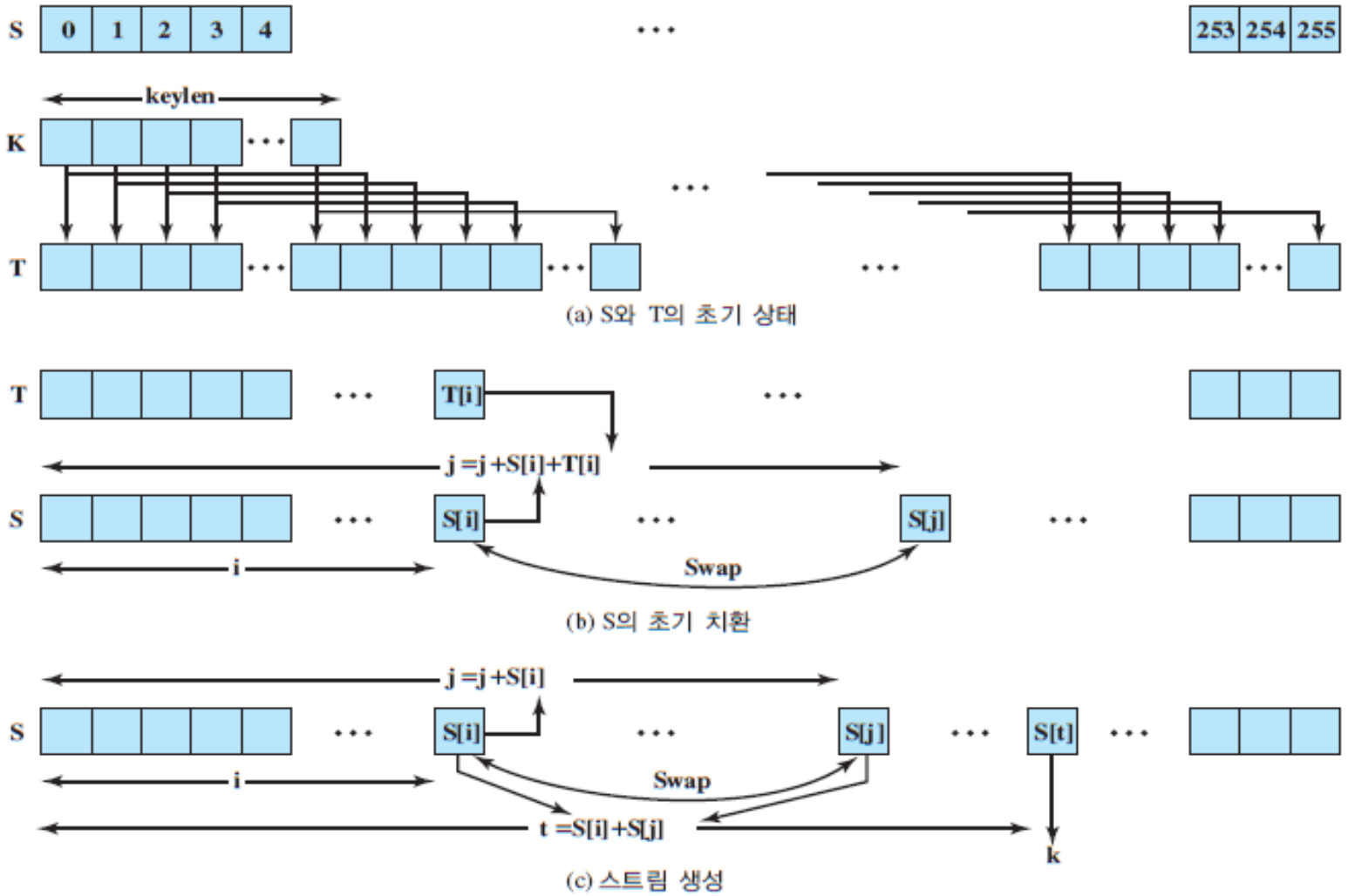
```
/* Initial Permutation of S */
j = 0;
for i = 0 to 255 do
j = (j + S[i] +T[i]) mod 256;
Swap (S[i], S[j]);
```

RC4 알고리즘 (4)

- RC 4에서 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];
```

RC4 알고리즘 (5)


S
0
1
2
3
4
253
254
255
keylen
K
T
(a) S와 T의 초기 상태
T
T[i]
j =j +S[i]+T[i]
S
S[i]
S[j]
i
Swap
(b) S의 초기 치환
j =j +S[i]
S
S[i]
S[j]
S[t]
i
Swap
t =S[i]+S[j]
k
(c) 스트림 생성

RC4 알고리즘 (6)

- RC4의 강도
  - RC4에 대한 실제적 공격 어려움
  - RC4를 이용한 WEP 프로토콜의 취약성
    - RC4에 입력으로 사용되는 키의 생성 방법에 문제