TRUYỀN VÀ BẢO MẬT THÔNG TIN

Bài 4:

Mã hóa đối xứng hiện đại (phần I)

VŨ THỊ TRÀ

Nội dung

- Bản tin, mã ASCII, biểu diễn nhị phân
- Mã dòng (Stream Cipher): A5/1 + RC4
- Mã khối (Block Cipher)
- Mã DES (Data Encryption Standard)
- Mã AES (Advanced Encryption Standard); Các mô hình ứng dụng của mã khối
- Một số thuộc tính hệ MHĐXHĐ; Trao đổi khóa bí mật bằng TT phân phối khóa KDC

Bản tin, mã ASCII, biểu diễn nhị phân

- Bản tin: **attack**
- Mã ASCII: **97 116 116 97 99 107**
- Biểu diễn nhị phân:

  01100001 01110100 01110100
  01100001 01100011 01101011

Ngôn ngữ chữ cái & hệ nhị phân

- Giả sử **bản rõ** là các chữ cái của một **ngôn ngữ** gồm có **8 chữ cái** A, B, C, D, E, F, G, H trong đó mỗi chữ cái được biểu diễn bằng **3 bít.**

Chữ cái	Nhị phân
A	000
B	001
C	010
D	011
E	100
F	101
G	110
H	111

Ngôn ngữ chữ cái & hệ nhị phân

- Giả sử dùng một **khóa K** 4 bít **0101** để mã hóa bản rõ là "**head**" bằng phép XOR:

Bản rõ:	**1111 0000 0011**	**(head)**
Khóa:	**0101 0101 0101**	
Bản mã:	**1010 0101 0110**	**(FBCG)**

→ Khóa lặp → Bộ sinh số ngẫu nhiên để tạo khóa dài, giả lập mã hóa One-Time Pad → Cơ sở của mã dòng (Stream cipher)

→ Khối được mã hóa bằng phép XOR với khóa → Không an toàn → Tìm kiếm các phép mã hóa phức tạp hơn XOR → Cở sở của mã khối (Block cipher)

Mã dòng

Mã dòng (Stream Cipher)

- **Các đặc tính**
- **Mô hình mã dòng**
- **Phương pháp mã dòng tiêu biểu**
  - **A5/1**: dùng trong mạng GSM
  - **RC4**: dùng trong giao thức SSL, hay WEP, WPA của mạng Wireless LAN

Nguyên lý của mã dòng

- Bản rõ được chia thành các đơn vị mã hóa:

$$P = p_0 p_1 p_2 \dots p_{n-1} \quad (p_i: k \text{ bít})$$

- Một bộ sinh dãy số ngẫu nhiên: dùng một khóa *K* ban đầu để sinh ra các số ngẫu nhiên có kích thước bằng kích thước đơn vị mã hóa

$$\textbf{StreamCipher (K)} \text{ --> } S = s_0 s_1 s_2 \dots s_{n-1} \quad (s_i: k \text{ bít})$$

- Mỗi số ngẫu nhiên được XOR với đơn vị mã hóa của bản rõ để có được bản mã:

$$c_0 = p_0 \oplus s_0 \, , c_1 = s_1 \oplus p_1, \dots , c_{n-1} = s_{n-1} \oplus p_{n-1};$$

$$C = c_0 c_1 c_2 \dots c_{n-1}$$

Ví dụ

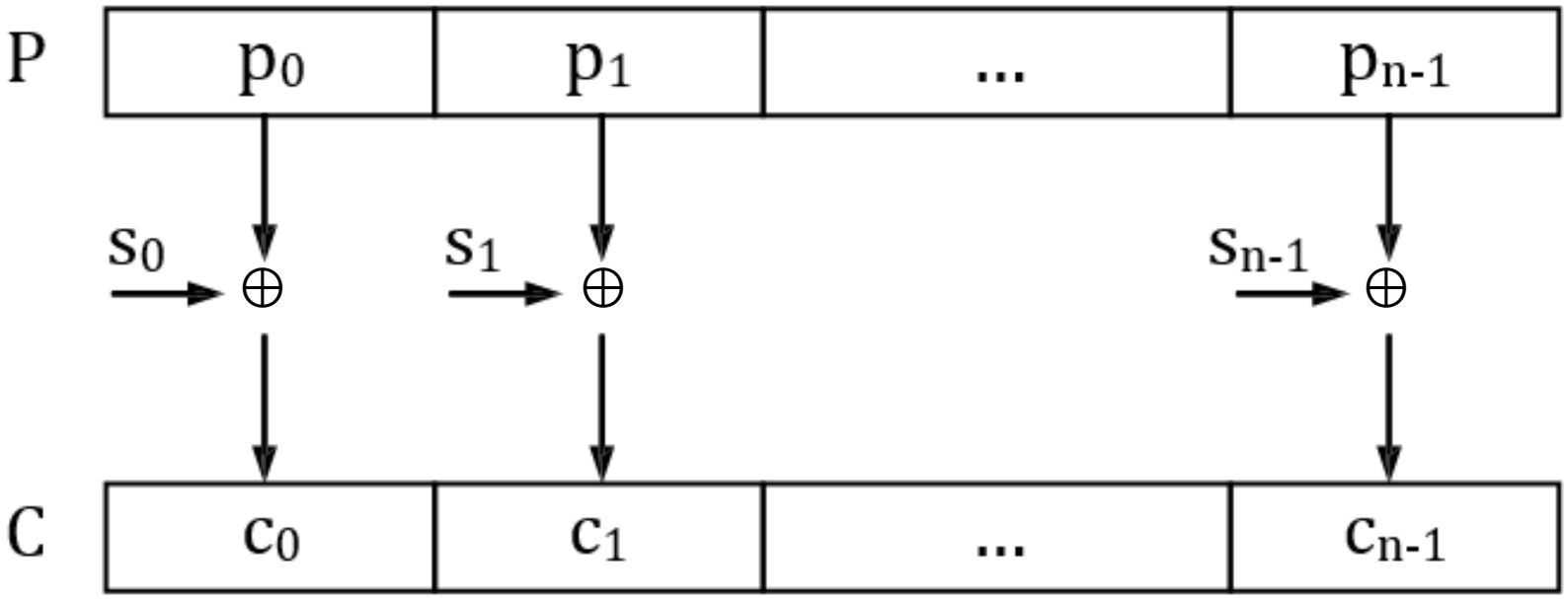
- Đơn vị mã hóa có chiều dài $k = 4$ bít, $n = 3$:

  $p_0 = 1111$, $p_1 = 0000$, $p_2 = 0011$ (head)

  $s_0 = 0101$, $s_1 = 1001$, $s_2 = 1010$

  $c_0 = 1010$, $c_1 = 1001$, $c_2 = 1001$ (FCDB)

Mô hình mã dòng



→ *s_i được sinh ra từ bộ sinh dãy số ngẫu nhiên StreamCipher (K)*

Nhận xét

- Mã hóa dòng tương tự như mã hóa Vigenere và mã hóa One-Time Pad. Điểm quan trọng nhất của các mã dòng là bộ sinh số ngẫu nhiên cho khóa đơn vị.
- Nếu chọn khóa có chiều dài ngắn như mã hóa Vigenere thì không bảo đảm an toàn, còn nếu chọn khóa có chiều dài bằng chiều dài bản tin như One-Time Pad thì lại không thực tế.
- Bộ sinh số của mã dòng cân bằng giữa hai điểm này, cho phép dùng một khóa ngắn nhưng phải bảo đảm một độ ngẫu nhiên cần thiết như khóa của One-time Pad

Mã dòng A5/1

A5/1:: Lịch sử và Ứng dụng

- A5/1 được sử dụng trong mạng thoại GSM ở châu Âu và Hoa Kỳ.
- A5/1 được phát triển vào năm 1987, khi GSM vẫn chưa được xem xét để sử dụng bên ngoài châu Âu, và A5/2 được phát triển vào năm 1989. Mặc dù cả hai đã bước đầu giữ bí mật, thiết kế tổng thể đã bị rò rỉ trong năm 1994 và các thuật toán đã hoàn toàn giải ngược thiết kế vào năm 1999 bởi Marc Briceno từ một điện thoại GSM.
- Năm 2000, khoảng 130 triệu khách hàng GSM tin A5/1 bảo vệ thông tin liên lạc bằng giọng nói của họ; vào năm 2011, nó là 4 tỷ.

https://en.wikipedia.org/wiki/A5/1

A5/1:: Nguyên tắc

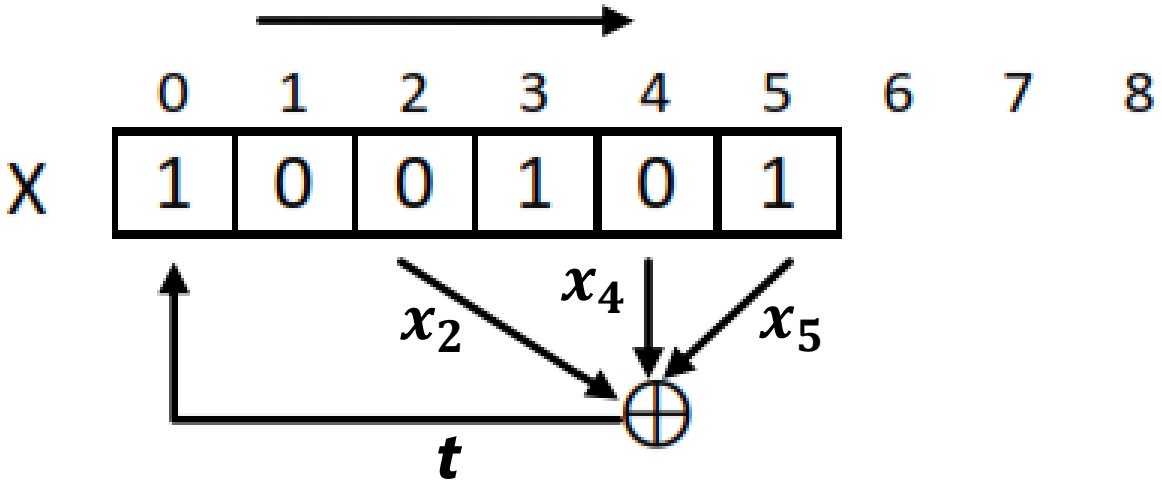
- Đơn vị mã hóa của A5/1 là 1 bít
- Bộ sinh số mỗi lần sẽ sinh ra hoặc bít 0 hoặc bít 1 để sử dụng trong phép XOR

Tiny A5/1

- Bộ sinh số gồm 3 thanh ghi *X, Y, Z*. Thanh ghi *X* gồm 6 bít, ký hiệu là ($x_0, x_1, \ldots, x_5$). Thanh ghi *Y* gồm 8 bít ($y_0, y_1, \ldots, y_7$). Thanh ghi *Z* lưu 9 bít ($z_0, z_1, \ldots, z_8$). Khóa *K* ban đầu có chiều dài 23 bít và lần lượt được phân bố vào các thanh ghi: $K \rightarrow XYZ$. Các thanh *X, Y, Z* biến đổi theo các nguyên tắc dưới đây:

Tiny A5/1:: Bộ sinh số:: Thanh X

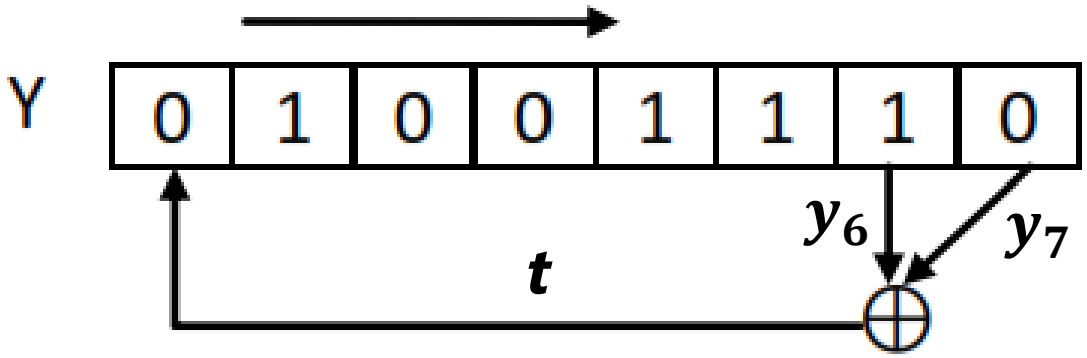
1) **Quay X:**



- $t = x_2 \oplus x_4 \oplus x_5$
- $x_j = x_{j-1}$ $với\ j = 5, 4, 3, 2, 1$
- $x_0 = t$

Tiny A5/1:: Bộ sinh số:: Thanh Y

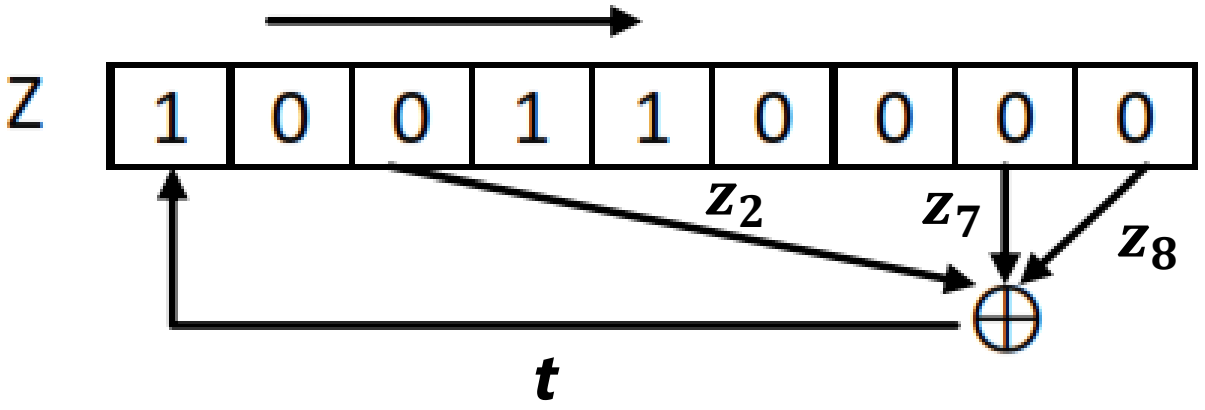
2) Quay Y:



- $t = y_6 \oplus y_7$
- $y_j = y_{j-1}$ $với j = 7, 6, 5, \ldots, 1$
- $y_0 = t$

Tiny A5/1:: Bộ sinh số:: Thanh Z

3) Quay Z:



- $t = z_2 \oplus z_7 \oplus z_8$
- $z_j = z_{j-1}$ $với\ j = 8, 7, 6, \ldots, 1$
- $z_0 = t$

Tiny A5/1:: Bộ sinh số:: Hàm *maj(x,y,z)*

- Cho 3 bít ***x, y, z***, ta định nghĩa một hàm ***maj(x, y, z)*** là hàm "**chiếm đa số**", nghĩa là nếu trong 3 bít *x, y, z* có từ hai bít 0 trở lên thì hàm trả về giá trị 0, nếu không hàm trả về giá trị 1.
- Tại bước sinh số thứ *i*, các phép tính sau được thực hiện:

$$\boldsymbol{m = maj(x_1, y_3, z_3)}$$

If $x_1 = m$ then *thực hiện quay X*

If $y_3 = m$ then *thực hiện quay Y*

If $z_3 = m$ then *thực hiện quay Z*

$$\rightarrow \boldsymbol{s_i = x_5 \oplus y_7 \oplus z_8}$$

Bít s_i được XOR với bít thứ i trong bản rõ để có được bít thứ i trong bản mã theo quy tắc của mã dòng.

Tiny A5/1:: Ví dụ 1

Mã hóa bản rõ **P=111** (chữ h) với khóa **K = 100101. 01001110.100110000.**

- Ban đầu gán giá trị của các thanh ghi X, Y, Z:

  X = 100101

  Y = 01001110

  Z = 100110000

- Bước 0: x_1= 0, y_3=0, z_3= 1 → m = 0 → quay X, Y

  X = 110010

  Y = 10100111 → $s_0 = 0 \oplus 1 \oplus 0 = 1$

  Z = 100110000

Tiny A5/1:: Ví dụ 1

- Bước 1: x_1= 1, y_3=0, z_3= 1 → m = 1 → quay X, Z

  X = 111001

  Y = 10100111 → $s_1 = 1 \oplus 1 \oplus 0 = 0$

  Z = 010011000

- Bước 2: x_1= 1, y_3=0, z_3= 0 → m = 0 → quay Y, Z

  X = 111001

  Y = 01010011 → $s_2 = 1 \oplus 1 \oplus 0 = 0$

  Z = 001001100

→ **Bản mã C=111 ⊕ 100 = 011** (chữ D)

Tiny A5/1:: Ví dụ 2

Nếu bản rõ **P=111.000** (chữ ha) với khóa **K = 100101.01001110.100110000.**

→ Tìm bản mã **C**?

A5/1:

Bộ sinh số A5/1 hoạt động giống như TinyA5/1. Kích thước thanh ghi X, Y, Z lần lượt là 19, 22 và 23 bít. Các bước quay X, Y, Z cụ thể là:

1) **Quay X**

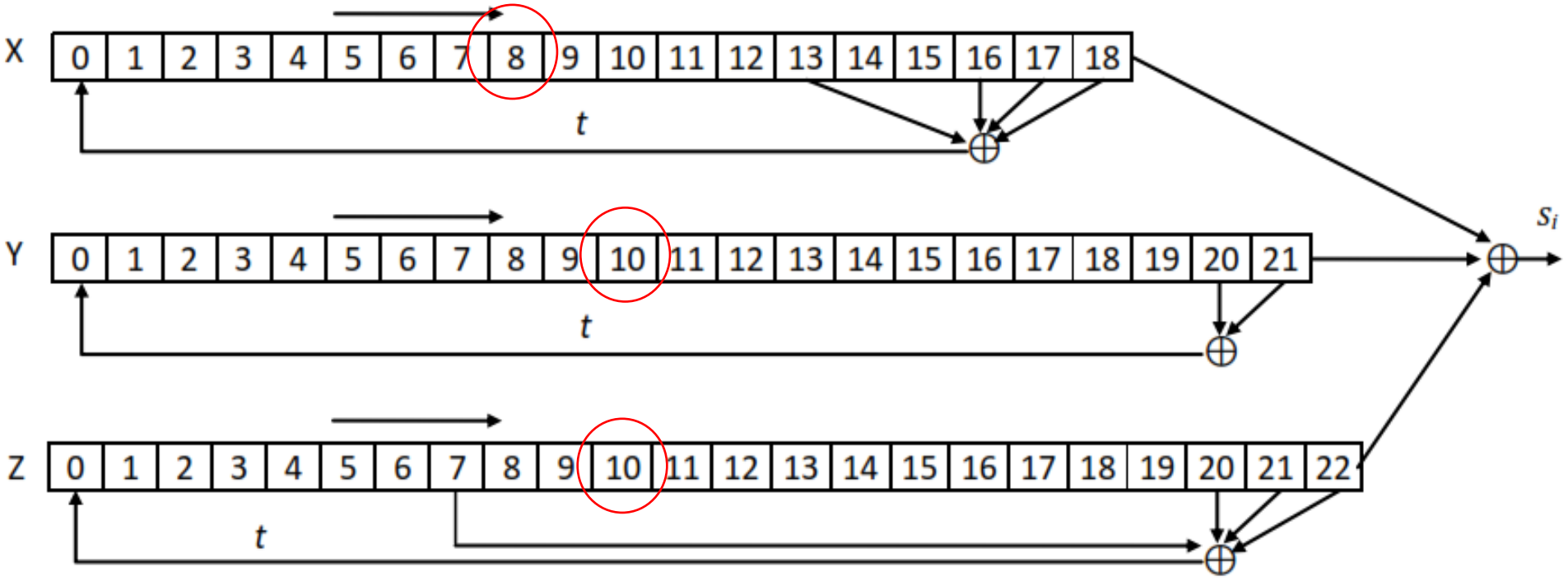
- $t = x_{13} \oplus x_{16} \oplus x_{17} \oplus x_{18}$
- $x_j = x_{j-1}$ $với\, j = 18, 17, 16, \ldots, 1$
- $x_0 = t$

A5/1

2) **Quay Y:**
   - $t = y_{20} \oplus y_{21}$
   - $y_j = y_{j-1}$ $với\, j = 21, 20, 19, \ldots, 1$
   - $y_0 = t$
3) **Quay Z:**
   - $t = z_7 \oplus z_{20} \oplus z_{21} \oplus z_{22}$
   - $z_j = z_{j-1}$ $với\, j = 22, 21, 20, \ldots, 1$
   - $z_0 = t$

A5/1

- Hàm *maj* được tính trên 3 bít x_8, y_{10}, z_{10}. Sau khi quay xong bít sinh ra là: $\boldsymbol{s_i = x_{18} \oplus y_{21} \oplus z_{22}}$
- Toàn bộ quá trình sinh dãy bít của A5/1



A5/1:: Nhận xét về công dụng

- Mã hóa A5/1 có thể được thực hiện dễ dàng trên các thiết bị phần cứng, tốc độ nhanh.
- Do đó, A5/1 đã từng được sử dụng để mã hóa các dữ liệu real-time như các dãy bít audio.
- Ngày nay, A5/1 được sử dụng để mã hóa dữ liệu cuộc gọi trong mạng điện thoại GSM.

Mã dòng RC4

Tiny RC4:: Nguyên lý

- Đơn vị mã hóa của TinyRC4 là 3 bít.
- TinyRC4 dùng 2 mảng S và T, mỗi mảng gồm 8 số nguyên 3 bít (từ 0 đến 7).
- Khóa là một dãy gồm N số nguyên 3 bít với N có thể lấy giá trị từ 1 đến 8.
- Bộ sinh số mỗi lần sinh ra 3 bít để sử dụng trong phép XOR. Quá trình sinh số của TinyRC4 gồm hai giai đoạn:

Tiny RC4:: Nguyên lý:: Giai đoạn khởi tạo

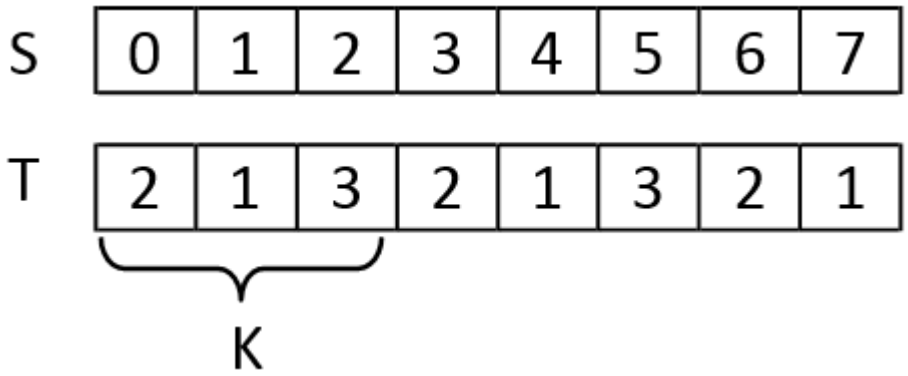
1) Giai đoạn khởi tạo:

```
/* Khoi tao day so S va T */
for  i = 0 to 7 do
     S[i] = i;
     T[i] = K[i mod N];
next i
/* Hoan vi day S */
j = 0;
for  i = 0 to 7 do
     j = (j + S[i] + T[i]) mod 8;
     Swap(S[i], S[j]);
next i
```

Tiny RC4:: Nguyên lý:: Giai đoạn khởi tạo

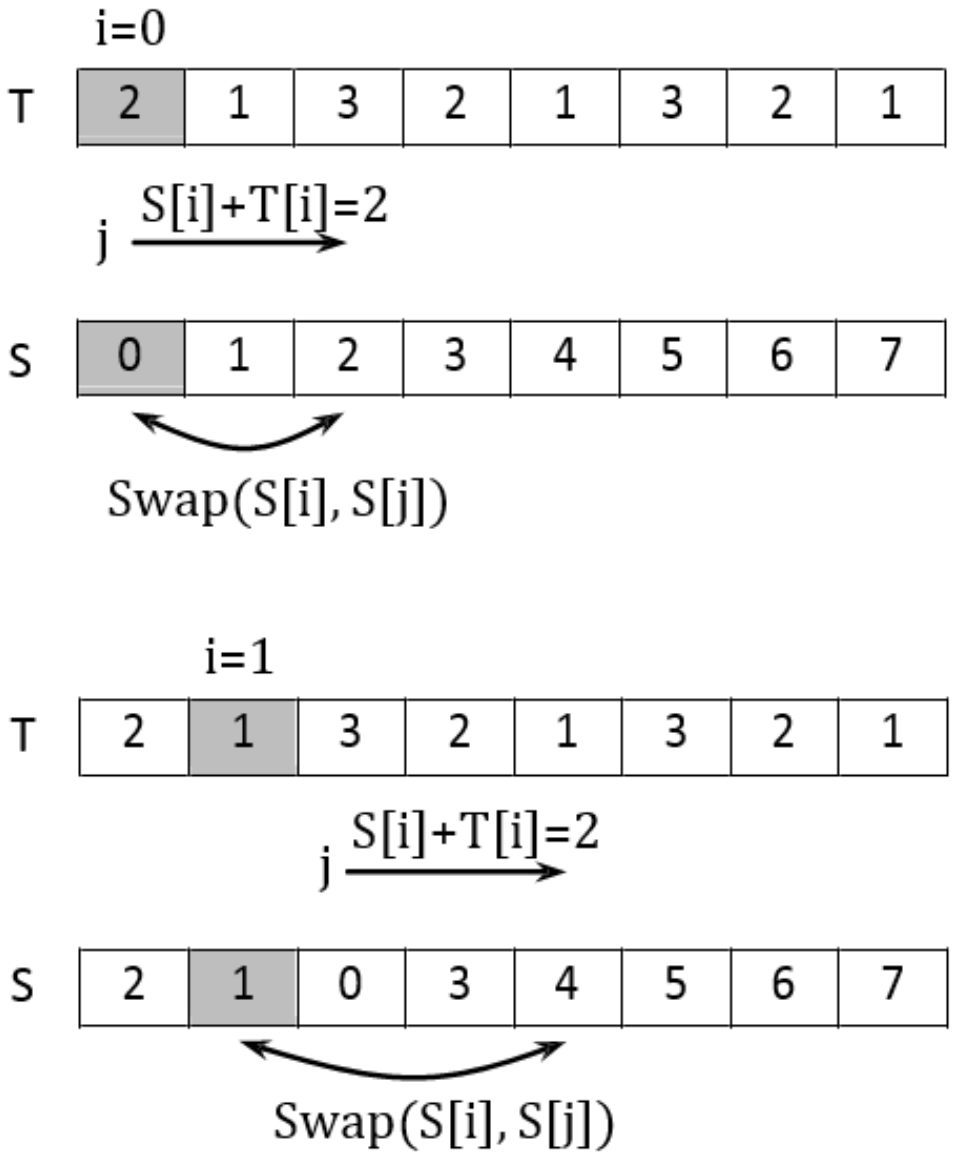
Ví dụ 1: mã hóa bản rõ P = 001000110 (từ „bag“) với khóa K gồm 3 số 2, 1, 3 (N=3)

- Khởi tạo S & T:



Tiny RC4:: Nguyên lý:: Giai đoạn khởi tạo

- Hoán vị S:



Tiny RC4:: Nguyên lý:: Giai đoạn khởi tạo

- Hoán vị S:

→ Quá trình thực hiện đến khi i=7 và lúc đó dãy S là 6 0 7 1 2 3 5 4

Tiny RC4:: Nguyên lý:: Giai đoạn sinh số

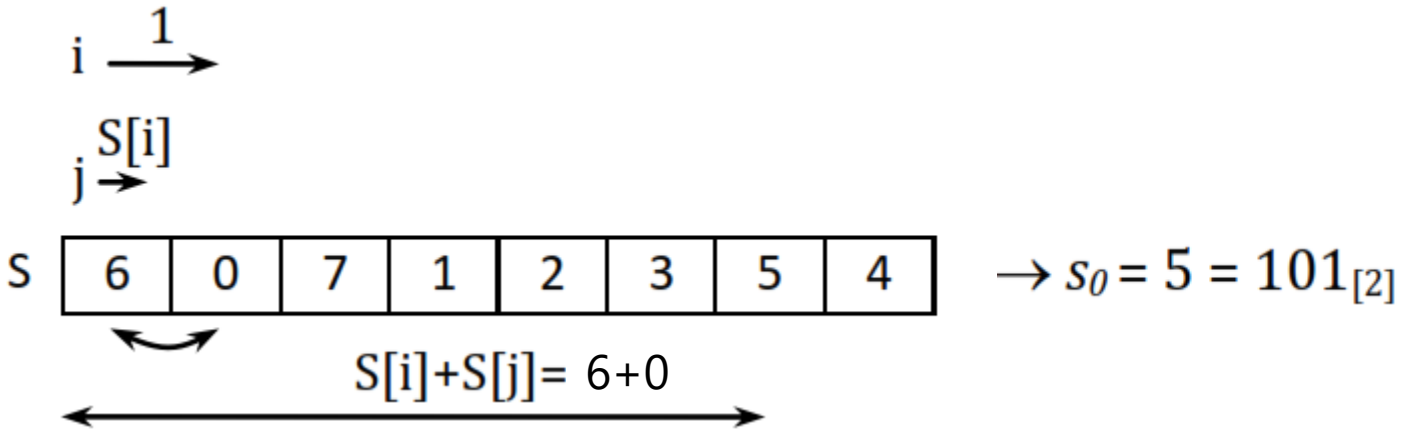
2) Giai đoạn sinh số:

```
i, j = 0;
while (true)
        i = (i + 1) mod 8;
        j = (j + S[i]) mod 8;
        Swap (S[i], S[j]);
        t = (S[i] + S[j]) mod 8;
        k = S[t];
end while;
```

Tiny RC4:: Nguyên lý:: Giai đoạn sinh số

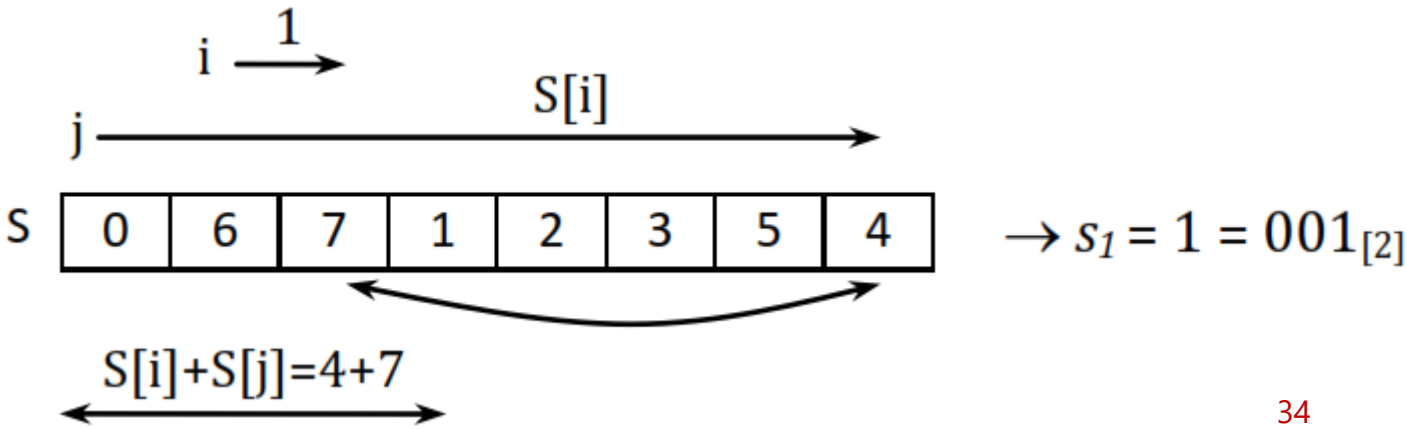
Tiếp tục ví dụ 1, quá trình sinh số mã hóa bản rõ "**bag**" với
S = 6 0 7 1 2 3 5 4

- Bước 0:



$\rightarrow s_0 = 5 = 101_{[2]}$

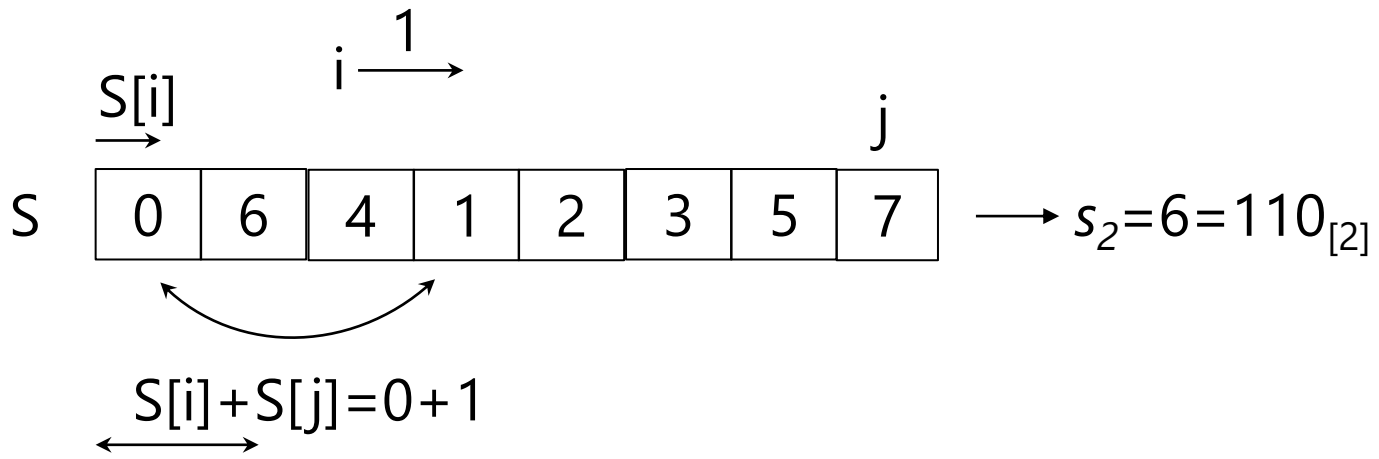
- Bước 1:



$\rightarrow s_1 = 1 = 001_{[2]}$

Tiny RC4:: Nguyên lý:: Giai đoạn sinh số

- Bước 2:



→ Vậy bản mã là **C = 001.000.110 ⊕ 101.001.110 = 100.001.000** (từ EBA)

RC4:: Nguyên tắc

- Đơn vị mã hóa của RC4 là 8 bít.
- Mảng S và T gồm 256 số nguyên 8 bít.
- Khóa là một dãy gồm N số nguyên 8 bít với N có thể lấy giá trị từ 1 đến 256.
- Bộ sinh số mỗi lần sinh ra 8 bít để sử dụng trong phép XOR. Quá trình sinh số của RC4 gồm hai giai đoạn:

RC4:: Nguyên tắc:: Giai đoạn 1

1) Giai đoạn khởi tạo:

```
/* Khoi tao day S va T*/
for  i = 0 to 255 do
     S[i] = i;
     T[i] = K[i mod N];
next i
/* Hoan vi day S */
j = 0;
for  i = 0 to 255 do
     j = (j + S[i] + T[i]) mod 256;
     Swap(S[i], S[j]);
next i
```

RC4:: Nguyên tắc:: Giai đoạn 2

2) Giai đoạn sinh số:

```
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];
end while;
```

RC4: Nhận xét

- Quá trình sinh số của RC4 cũng sinh ra dãy số ngẫu nhiên, khó đoán trước, vì vậy RC4 đạt được mức độ an toàn cao theo tinh thần của mã hóa One-Time Pad.
- Mã hóa RC4 hoàn toàn được thực hiện trên các số nguyên một byte do đó tối ưu cho việc thiết lập bằng phần mềm và tốc độ thực hiện nhanh.

RC4: Ứng dụng

- RC4 dùng trao giao thức SSL để bảo mật dữ liệu trong quá trình truyền dữ liệu giữa Web Server và trình duyệt Web.
- RC4 được sử dụng trong mã hóa WEP của mạng Wireless LAN.

Q&A

1. Mã dòng giống và khác với mã Vigenere và mã One-Time Pad ở điểm gì?
2. Mã dòng A5/1 dùng đơn vị mã hóa có kích thước là bao nhiêu?