* Simplified RC4 excampe :→

① suppose an S-array of length 8

S = [0, 1, 2, 3, 4, 5, 6, 7, 8]

② suppose key is

K = [1 2 3 6]

③ Plain text

P = [1 2 2 2]

T = [1 2 3 6 1 2 3 6]

```
j = 0
for i = 0 to 7 do
    j = j + S[i] + T[i] mod 8
    swap S[i] and S[j]
end
```

```
for i = 0
    j = (0 + 0 + 1) mod 8 = 1
    swap (S[0], S[1])
```

S = [1 0 2 3 4 5 6 7]

for i = 1

j = j + S[i] + T[i] mod 8

swap S[1], S[3]

S = [1 3 2 0 4 5 6 7]

for i = 2

j = j + S[i] + T[i] mod 8

= (3 + 2 + 3) mod 8 = 0

swap S[2], S[0]

S = [2 3 1 0 4 5 6 7]

for i = 3

j = j + S[i] + T[i] mod 8

= (0 + 0 + 6) mod 8 = 6

swap S[3], S[6]

S = [2 3 1 6 4 5 0 7]

for i = 4

j = j + S[i] + T[i] mod 8

= (6 + 4 + 1) mod 8 = 3

swap S[4], S[3]

S = [2 3 1 4 6 5 0 7]

for i = 5

$j = j + S[i] + T[i] \bmod 8.$

$= (3 + 5 + 2) \bmod 8 = 2$

swap S[5], S[2]

S = [2 3 5 4 6 1 0 7]

for i = 6

$j = (2 + 0 + 3) \bmod 8 = 5$

swap S[6], S[5]

S = [2 3 5 4 6 0 1 7]

for i = 7

$j = (5 + 7 + 6) \bmod 8 = 2$

swap S[7] S[2]

S = [2 3 7 4 6 0 1 5]

* simplified stream generation :-

```
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]; }
```

first Iteration:→

i = (0+1) mod 8 = 1
j = (0 + S[i]) mod 8
=) (0 + 3) mod 8 = 3
swap S[1], S[3]

S = [2 4 7 3 6 0 1 5]
t = (S[1] + S[3]) mod 8
=) (4 + 3) mod 8 = 7
k = S[7] = 5

2nd Iteration:→

S = [2 4 7 3 6 0 1 5]
i = (1+1) mod 8 = 2
j = (3 + S[2]) mod 8
=) (3+7) mod 8 = 2
swap S[2], S[2]

S = [2 4 7 3 6 0 1 5]

t = (S[2] + S[2] mod 8
=) (7 + 7) mod 8 = 6
k = S[6] = 1

3rd Iteration: →

S = [2 4 7 3 6 0 1 5]

i = (2+1) mod 8 = 3

j = (2+S[3]) mod 8 => (2+3) mod 8 = 5

swap S[3], S[5]

S = [2 4 7 0 6 3 1 5]

t = (S[3]+S[5]) mod 8 =>

(0+3) mod 8 = 3

K = S[3] = 0

4th Iteration: →

S = [2 4 7 0 6 3 1 5]

i = (3+1) mod 8 = 4

j = (5+S[4]) mod 8 => 3

swap S[4], S[3]

S = [2 4 7 6 0 3 1 5]

t = (S[4]+S[3]) mod 8 =>

(0+6) mod 8 = 6

K = S[6] = 1

Encryption :-

KS = [5 1 0 1]
PT = [1 2 2 2]
Ct = PT XOR KS

Pt = 0001 0010 0010 0010
kS = 0101 0001 0000 0001
Ct = 0100 0011 0010 0011
C.t. = 4 3 2 3

Decryption :-

Pt = Ct XOR XS

CT = 0100 0011 0010 0011
KS = 0101 0001 0000 0001
PT = 1 2 2 2