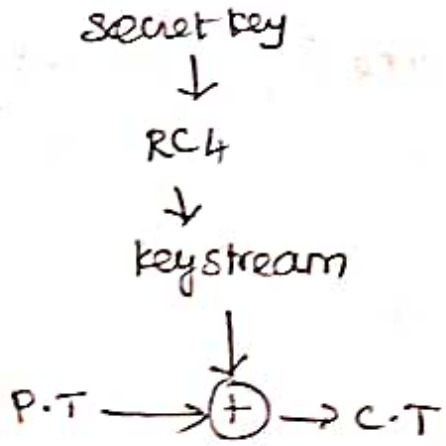
RC4 - Rivest cipher 4

RC4 - Stream cipher. variable length key algorithm. encrypts byte by byte at a time. Pseudo random bit generator and key scheduling algorithm is used.



Algorithm

→ user selected random key.

→ variable key from 1 to 256 bytes.

code for array initialization

```
char S[256];
int i;
for(i=0; i<256; i++)
  S[i] = i
```

code for key scheduling algorithm

```
int i, j = 0;
for(i=0; i<256; i++)
{
  j = (j + S[i] + S[j]) mod 256;
  swap(S[i], S[j]);
}
```

Code for PRGA . Pseudo Random generation Algorithm

```
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];
}
```

Ex: P[1 2 2 2] K = [1 2 3 6]

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

T = [1 2 3 6 1 2 3 6]. temporary vector

KSA - Key scheduling Algorithm.

```
i = 0 to 7
j = (j + S[i] + S[j]) mod 8
swap S[i], S[j].
```

i=0 j=0

j = [0 + 0 + 1] mod 8

j = 1 mod 8 j=1

swap [s[i], s[j]]

swap (s[0], s[1])

0 1 2 3 4 5 6 7

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

i=1, j=1 (prev j value)

j = [1 + 0 + 2] mod 8

j = 3 mod 8

j = 3

swap [s[1], s[3])

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

final

s = [2, 3, 7, 4, 6, 0, 5]

PRGA:

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]

i=0 j=0

i = (0+1) mod 8

i = 1 mod 8

i = 1

T = 0 1 2
1, 2, 3,

j = [0 + s[1]] mod 8

j = [0 + 3] mod 8

j = 3.

swap s[i], s[j];

swap s[1] s[3]

s = [2, 4, 7, 3, 6, 0, 5]

$t = (S[1] + S[3]) \bmod 8$

$t = [4^1 + 3] \bmod 8$

$t = 7$

$K = S[7]$

$K = 6$

6 XOR = 1

$110 \oplus 001 =$

$= 111 = (7)$

$\oplus$ TRUE $= 0$

$1 + 1 = 0$

$0 \oplus 0 = 0$

$1 + 0 = 1$

$0 + 1 = 1$