Nama : Auliya Rahman Asdar
NIM : E1E120025
Mata Kuliah : Kriptografi

Key- Scheduling Algorithm (KSA)

Diketahui :
Key (k) : saputra1 Len(k) = 8
Array S : [0, 1, 2, 3, ..., 254, 255]

1.) iterasi Pertama
i = 0
j = 0

j = (j + S[i] + k[i mod len(k)]) mod 256
j = (0 + 0 + k[0 mod 8]) mod 256
j = (k[0]) mod 256
j = ("s") mod 256
j = 115 mod 256
j = 115
Swap = (S[i], S[j])
= (S[0], S[115])
Array S = [115, 1, 2, 3, 4, 5, 6, ..., 114, 0, 116, ..., 254, 255]

2) iterasi kedua
i = 1
j = 115

j = (j + S[i] + k[i mod len(k)]) mod 256
= (115 + 1 + k[1 mod 8]) mod 256
= (116 + k[1]) mod 256
= (116 + "a") mod 256
= (116 + 97) mod 256
= 213 mod 256
j = 213
Swap: (S[i], S[j])
= (S[1], S[213])
Array S = [115, 213, 2, 3, ..., 114, 0, 116, ..., 212, 1, 213, ..., 254, 255]

3.) iterasi ketiga

i = 2

j = 213

j = (j + S[i] + k[i mod len(k)]) mod 256

j = (213 + 2 + k[3 mod 8]) mod 256

j = (215 + k[3]) mod 256

= (215 + "p") mod 256

= (215 + 112) mod 256

= 327 mod 256

j = 71

Swap = (S[i], S[j])

= (S[2], S[71])

Array S = [115, 213, 71, 3, ..., 70, 2, 72, ..., 114, 0, 116, ..., 212, 1, 214, ..., 254, 255]

4) iterasi keempat

i = 3

j = 71

j = (j + S[i] + k[i mod len(k)]) mod 256

= (71 + 3 + k[3 mod 8]) mod 256

= (74 + k[3]) mod 256

= (74 + "u") mod 256

= (74 + 117) mod 256

= 191 mod 256

j = 191

Swap = (S[i], S[j])

= (S[3], S[191])

Array S = [115, 213, 71, 191, 4, ..., 70, 2, 72, ..., 114, 0, 116, ..., 190, 3, 192, ..., 212, 1, 214, ..., 254, 255]

5). iterasi kelima

i = 4

j = 191

j = (j + S[i] + k[i mod len(k)]) mod 256

= (191 + 4 + k[4 mod 8]) mod 256

= (195 + k[4]) mod 256

= (195 + "t") mod 256
= (195 + 116) mod 256
= 311 mod 256
j = 55
swap = (S[i], S[j])
= (S[4], S[55])
Array S = [115, 213, 71, 191, 55, 5, ..., 54, 4, 56, ..., 70, 2, 72, ..., 114, 0,
116, ..., 190, 3, 192, ... 212, 1, 214, ..., 254, 255]

6) iterasi keenam
i = 5
j = 55

j = (j + S[i] + k [i mod len(k)]) mod 256
= (55 + 5 + k[5 mod 8]) mod 256
= (60 + k[5]) mod 256
= (60 + "r") mod 256
= (60 + 114) mod 256
= 174 mod 256
j = 174
swap = (S[i], S[j])
= (S[5], S[174])
Array S = [115, 213, 71, 191, 55, 174, 6, ..., 54, 4, 56, ..., 70, 2, 72, ...,
114, 0, 116, ..., 173, 5, 175, ..., 190, 3, 192, ... 212, 1, 214, ...
254, 255]

7) iterasi ketujuh
i = 6
j = 174

j = (j + S[i] + k [i mod len(k)]) mod 256
= (174 + 6 + k [6 mod 8]) mod 256
= (180 + k [6]) mod 256
= (180 + "a") mod 256
= (180 + 97) mod 256
= 277 mod 256
j = 21
swap = (S[i], S[j])
= (S[6], S[21])

Array S = [115 213, 71, 191, 55, 174, 21, 7, ..., 20, 6, 22, ..., 54, 4, 56, ...,
70, 2, 72, ..., 114, 0, 116, ..., 173, 5, 175, ..., 212, 1, 214, ...
254 255]

8.) Iterasi kedelapan
i = 7
j = 21

j = (j + S[i] + K[i mod len(k)]) mod 256
= (21 + 7 + k[7 mod 8]) mod 256
= (28 + k[7]) mod 256
= (28 + "1") mod 256
= (28 + 49) mod 256
= 77 mod 256
j = 77
Swap = (S[i], S[j])
= (S[7], S[77])
Array S = [115, 213, 71, 191, 55, 174, 21, 77, 8, ..., 20, 6, 22, ..., 54, 4, 56, ..., 70, 2,
...., 76, 7, 78, ..., 114, 0, 116, ..., 173, 5, 175, ..., 212, 1, 214, ...,
254, 255]

Psaudo - Random Generation Automaton (PRGA)

1.) Iterasi pertama
i = 0
j = 0

i = (i+1) mod 256
= (0 + 1) mod 256
= 1 mod 256
i = 1

j = (j + S[i]) mod 256
= (0 + S[1]) mod 256
= (0 + 213) mod 256
= 213 mod 256
j = 213

Swap = (S[i], S[j])

= (S[i], S[213])

Array S = [115, 1, 71, 191, 55, 174, 21, 77, ..., 20, 6, 22, ..., 54, 4, 56, ..., 76, 7, 78, ..., 114, 0, 116, ..., 173, 5, 175, ..., 190, 3, 191, ..., 254, 255]

t = (S[i] + S[j]) mod 256
= (S[1] + S[213]) mod 256
= (1 + 213) mod 256
= 214 mod 256
t = 214

K = S[t] = 214 P = 2 = 00110010
C = K ⊕ P
214 = 11010110 C = ä, atau desimal 228
2 = 00110010
228 = 11100100 ⊕

2) iterasi kedua
i = 1
j = 213

i = (i+1) mod 256
i = (1+1) mod 256
= 2 mod 256
= 2

j = (j + S[i]) mod 256
= (213 + S[2]) mod 256
= (213 + 71) mod 256
= 284 mod 256
j = 28

swap = (S[i], S[j])
= (S[2], S[28])

Array S = [115, 1, 28, 191, 55, 174, 21, 77, 8, ..., 20, 6, 22, ..., 27, 71, 29, 54, 4, 56, ..., 70, 2, 72, ..., 76, 7, 78, ..., 114, 0, 116, ..., 173, 5, 175, ..., 190, 3, 192, ..., 254, 255]

t = (S[i] + S[j]) mod 256
= (S[2] + S[28]) mod 256
= (28 + 71) mod 256
= 99 mod 256
t = 99
P = @ = 00110000
K = S[t] = 99 = 01100011
C = K ⊕ P

99 = 0 11 000 11
0 01 100 00 ⊕
83 = 0 10 100 11

C = "S" atau desimal 83

3.) iterasi ketiga
i = 2
j = 28

i = (i+1) mod 256
= (2+1) mod 256
= 3 mod 256
i = 3

j = (j + S[i]) mod 256
= (28 + S[3]) mod 256
= (28 + 191) mod 256
= 219 mod 256
j = 219

Swap = (S[i], S[j])
= (S[3], S[219])

Array S = [115, 1, 28, 219, 55, 174, 21, FF, 8, ..., 20, 6, 22, ..., 27, 71, 29, ...,
54, 4, 56, ..., 70, 2, 72, ..., 76, 7, 78, ..., 114, 0, 116, ...,
173, 5, 175, ..., 190, 3, 192, ..., 218, 191, 220, ... 254, 255]

t = (S[i] + S[j]) mod 256
= (S[3] + S[219]) mod 256
= (219 + 191) mod 256
= 154

K = S[t] = 154 = 10011010
P = 2 = 00110010
C = K ⊕ P

154 = 10011010

168 = 10101000 ⊕ 00110010

C = 11 atau decimal 168

9.) iterasi keempat

i = 3

j = 219

i = (i+1) mod 256

= (3+1) mod 256

= 4 mod 256

i = 4

j = (j + S[i]) mod 256

= (219 + S[4]) mod 256

= (219 + 55) mod 256

= 274 mod 256

= 18

swap = (S[i], S[j])

= (S[4], S[18])

Array S = [115, 1, 20, 219, 18, 174, 21, 77, 0, ..., 17, 55, 19, ..., 20, 6, 22, ...,
27, 71, 29, ..., 54, 4, 56, ..., 70, 2, 72, ..., 76, 7, 78,
...., 119, 0, 116, ..., 173, 5, 175, ..., 190, 3, 191, ..., 218,
191, 220, ..., 254, 255]

t = (S[i] + S[j]) mod 256

= (S[4] + S[18]) mod 256

= (18 + 55) mod 256

= 73.

k = S[t] = 73 = 01001001

P = 5 = 00110101

C = K ⊕ P

73 = 01001001

00110101 ⊕

124 = 01111100

C = | atau decimal 124