KRIPTOGRAFI

Nama : A'tika Nurfadilah
NIM : E1E120001

RC4

➤ Key-Scheduling Algorithm (KSA)

↳ Kunci : saputra1

↳ Array S: [0,1,2,3,4,5,..., 100, 101, 102, 103, 104, 105, ..., 251, 252, 253, 254, 255]

➤ Iterasi: pertama

i = 0, j = 0

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

= (0 + 0 + k[0 % 8]) % 256

= (k[0]) % 28 => ("s") % 256 ⇒ 115 % 256 = 115 //

swap (S[i], S[j]) => swap (S[0], S[115])

Array S : [115, 1, 2, 3, 4, ..., 112, 113, 114, 0, 116, 117, ..., 251, 252, 253, 254, 255]

➤ Iterasi: kedua

i = 1, j = 115

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

= (115 + S[1] + k[1 % 8]) % 256

= (115 + 1 + k[1]) % 256 => (116 + ("a")) % 256

= (116 + 97) % 256 => (213) % 256 = 213 //

swap (S[i], S[j]) ⇒ swap (S[1], S[213])

Array S = [115, 213, 2, 3, 4, ..., 114, 0, 116, 117, ..., 211, 212, 1, 214, ..., 251, 252, 253, 254, 255]

➤ Iterasi: ketiga

i = 2, j = 213

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

= (213 + S[2] + k[2 % 8]) % 256

= (213 + 2 + k[2]) % 256

= (213 + 2 + "p") % 256 ⇒ (215 + 112) % 256 ⇒ 327 % 256 = 327

= 71 //

swap (S[i], S[j]) => swap (S[2], S[71])

Array S = [115, 213, 71, 3, 4, 5, ..., 69, 70, 2, 72, ..., 114, 0, 116, 117, ..., 211, 212

1, 214, ..., 251, 252, 253, 254, 255]

→ Iterasi keempat

i = 3 , j = 71

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

= (71 + S[3] + k [3 % 8]) % 256

= (71 + 3 + k[3]) % 256

= (74 + "u") % 256 => (74 + 117) % 256 => 191 % 256 => 191 //

swap (S[i], S[j]) => swap (s[3], s[191])

Array S = [115, 213, 71, 191, 4, 5, ... 69, 70, 2, 72, ..., 114, 0, 116, 117, ..., 190, 3, 192, 193, ...,
211, 212, 1, 214, ..., 251, 252, 253, 254, 255]

→ Iterasi kelima

i = 4 , j = 191

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

= (191 + S [4] + k [4 % 8]) % 256

= (191 + 4 + k [4]) % 256

= (195 + "t") % 256 => (195 + 116) % 256 => 311 % 256 => 55 //

swap (S[i], S[j]) => swap [S[4], S[55])

Array S = [115, 213, 71, 191, 55, 5, ..., 54, 4, 56, 57, ..., 69, 70, 2, 72, ..., 114, 0, 116, 117, ...,
190, 3, 192, 193, ..., 211, 212, 1, 214, ..., 251, 252, 253, 254, 255]

→ Iterasi keenam

i = 5 , j = 55

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

= (55 + S [5] + k [5 % 8]) % 256

= (55 + 5 + k [5]) % 256

= (55 + 5 + "r") % 256 => (60 + 114) % 256 => 174 % 256 => 174 //

swap (S[i], S[j]) => swap (S [5] , S [174])

Array S = [115, 213, 71, 191, 55, 174, 6, 7, 8, ..., 54, 5, 56, 57, ..., 69, 70, 2, 72, ...,
114, 0, 116, 117, ..., 172, 173, 5, 175, ..., 190, 3, 192, 193, ..., 211, 212, 1, 214, ...,
251, 252, 253, 254, 255]

→ Iteras: ketujuh

i = 6 , j = 174

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

= (174 + S [6] + k [6 % 8]) % 256

= (174 + 6 + k [6]) % 256

= (180 + "a") % 256 => (180 + 97) % 256 => 277 % 256 = 21 //

swap (S [i], S [j]) => swap (S [6], S [21])

Array S = [115, 213, 71, 191, 55, 174, 21, 7, 8, ..., 19, 20, 6, 22, 23, ..., 54, 4, 56, 57, ..., 69,
70, 2, 72, ..., 114, 0, 116, 117, ..., 172, 173, 5, 175, ..., 190, 3, 192, 193, ..., 211, 212,
1, 214, ..., 251, 252, 253, 254, 255]

• Iterasi kedelapan

i = 7, j = 21

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

= (21 + S[7] + k[7 % 8]) % 256

= (21 + 7 + k[7]) % 256

= (28 + "1") % 256 => (28 + 49) % 256 => 77 % 256 => 77 //

swap (S[i], S[j]) => swap (S[7], S[77])

Array S = [115, 213, 71, 191, 55, 21, 77, 8, ..., 19, 20, 6, 22, 23, ..., 54, 4, 56, 57, ..., 69, 70, 2, 72, ..., 76, 7, 78, ..., 114, 0, 116, 117, ..., 172, 173, 5, 175, ..., 190, 3, 192, 193, ..., 211, 212, 1, 214, ..., 251, 252, 255, 254, 255]

Pseudo - Random Generation Algorithm (PRGA)

↳ Plainteks = 2001

• Iterasi pertama (P[0])

i = 0, j = 0

i = (i + 1) mod 256 => (0+1) mod 256 => 1 % 256 = 1 //

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

swap (S[i], S[j]) => swap (S[1], S[213])

Array S = [115, 1, 71, 191, 55, 174, 21, 77, 8, ..., 20, 6, 22, 23, ..., 54, 4, 56, 57, ..., 70, 2, 72, 73, ..., 76, 7, 78, 79, ..., 114, 0, 116, 117, ..., 173, 5, 175, 176, ..., 190, 3, 192, 193, ..., 212, 213, 214, ..., 251, 252, 253, 254, 255]

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

= (S[1] + S[213]) % 256 => (1 + 213) % 256 => 214 % 256 => 214

u = S[t] = S[214] = 214 => 11010110

c = u ⊕ P[0] => u ⊕ 2 => 11010110 ⊕ 00110010

= 11010110

00110010 ⊕

11100100 => 228 => ä

⇒ Iterasi kedua (P[1])

i = 1, j = 213

i = (i+1) mod 256 ⇒ (1+1) mod 256 ⇒ 2 % 256 = 2 //

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

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

Array S = [115, 1, 28, 191, 55, 174, 21, 77, 8, ... 19, 20, 6, 22, 23, ..., 26, 27, 71, 29, 30, ...,
53, 54, 4, 56, 57, ..., 69, 70, 2, 73, ..., 76, 7, 78, ..., 114, 0, 116, 117, ...,
172, 173, 5, 175, ..., 190, 3, 192, 193, ..., 212, 213, 214, 215, ..., 251, 252,
253, 254, 255]

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

= (S[1] + S[28]) % 256 ⇒ (28 + 71) % 256 ⇒ 99 % 256 ⇒ 99 //

u = S[t] ⇒ S[99] = 99 ⇒ 1100011

c = u ⊕ P[1] ⇒ u ⊕ 0 ⇒ 1100011 ⊕ 110000

= 1100011

0110000 ⊕

1010011 ⇒ 83 ⇒ S

⇒ Iteras: ketiga (P[2])

i = 2, j = 28

i = (i+1) mod 256 ⇒ (2+1) % 256 ⇒ 3 % 256 = 3 //

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

swap (S[i], S[j]) ⇒ ~~swap (S[3], S[219])~~ swap (S[3], S[219])

Array S = [115, 1, 28, 219, 55, 174, 21, 77, 8, ..., 218, 191, 220, ..., 255]

t = (S[i] + S[j]) % 256 ⇒ (S[3] + S[219]) % 256 ⇒ (219 + 191) % 256 ⇒ 154 //

u = S[t] = S[154] ⇒ 154 ⇒ 10011010

c = u ⊕ P[2] ⇒ u ⊕ 0 ⇒ 10011010 ⊕ 00110000 ⇒ 10101010 ⇒ 170 ⇒ ª

⇒ Iterasi keempat (P[3])

i = 3, j = 219

i = (i+1) mod 256 ⇒ (3+1) % 256 ⇒ 4 % 256 = 4

j = (j + S[i]) % 256 ⇒ (219 + S[4]) % 256 ⇒ (219 + 55) % 256 ⇒ (274 % 256) ⇒ 18 //

swap (S[i], S[j]) ⇒ swap (S[4], S[18])

Array S = [115, 1, 28, 219, 18, 174, 21, 77, 8, ..., 17, 55, 19, 20, ..., 255]

t = (S[i] + S[j]) % 256 ⇒ (S[4], S[18]) % 256 ⇒ (18 + 55) % 256 ⇒ 73 //

u = S[t] = S[73] = 73 ⇒ 1001001

c = u ⊕ P[3] ⇒ u ⊕ 1 ⇒ 1001001 ⊕ 0110001 ⇒ 1111000 ⇒ 120 ⇒ x

Plainteks : 2001

Chiperteks : äSªx ⇒ Desimal : 228 83 170 120