Nama : Syarifah Nurhafni A.
NIM : E1E1 20 099
Kelas : Genap

Tugas kriptografi

Algoritma RC4
1. KSA 2. PRGA

1. K = Saputra => $K_0 = s, K_1 = a, K_2 = p, K_3 = u, K_4 = t, K_5 = r, K_6 = a, K_7 = 1$

array S = [0, 1, 2, 3, 4, 5, 6, ..., 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255]

~~j = 0~~ ~~i = 0~~

iterasi Pertama -> i = 0

~~K = j = (j+s)~~ j = 0

-> j = (j + S(i) + K [i mod len(k)]) mod 256
= (0 + 0 + k [0 % 8]) % 256
= (k [0]) % 256
= ("S") % 256 => nilai desimal dari "S" = 115
= 115 % 256

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

Array S = [115, 1, 2, 3, 4, 5, 6, 7, ..., 110, 111, 112, 113, 114, 0, 116, 117, ..., 199, 200, 201, 202, 203, 204, 205, ..., 250, 251, 252, 253, 254, 255]

iterasi kedua ⟶ i = 1

j = 115

=> j = (j + S(i) + k [i % len(k)]) % 256
= (115 + S(1) + k [1 % 8]) % 256
= (115 + 1 + k [1]) % 256
= (116 + "a") % 256 => desimal dari "a" = 97
= (116 + 97) % 256
= 213 % 256

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

Array S = [115, 213, 2, 3, 4, 5, 6, 7, ..., 112, 113, 114, 0, 116, ..., 210, 211, 212, 1, 214, ..., 250, 251, 252, 253, 254, 255]

Iterasi Ketiga → i = 2

j = 213

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

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

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

= (215 + "p") % 256 ⇒ desimal dari "p" = 112

= (215 + 112) % 256

= 327 % 256

j = 71

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

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

Array S = [115, 213, 71, 3, 4, 5, 6, 7, ..., 69, 70, 2, 72, ..., 112, 113, 114, 0,
116, ..., 210, 211, 212, 1, 214, ..., 250, 251, 252, 253, 254, 255]

Iterasi keempat → i = 3

j = 71

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

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

= ~~(79 + "u") % 256~~ (71 + 3 + k[3]) % 256

= (74 + "u") % 256 ⇒ desimal dari "u" = 117

= (74 + 117) % 256

= 191 % 256

j = 191

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

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

array S = [115, 213, 71, 191, 4, 5, 6, 7, ..., 69, 70, 2, 72, ..., 112, 113, 114, 0,
116, ..., 189, 190, 3, 192, ..., 210, 211, 1, 214, ..., 250, 251,
252, 253, 254, 255]

Iterasi kelima → i = 4

j = 191

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

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

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

= (195 + 116) % 256

= 311 % 256

j = 55

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

Swap (S [a] , S [55]

Array S : [115, 213, 71, 191, 55, 5, 6, 7, 8, ..., 53, 59, 4, 56, 57, ..., 69,
70, 2, 72, 73, ..., 113, 114, 0, 116, 117, ..., 189, 190, 3, 192, ...
211, 212, 1, 214, ..., 250, 251, 252, 253, 254, 255]

Iterasi keenam ——> i = 5

j = 55

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

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

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

= (60 + "r") mod 256 -> ascii r = 119 (desimal)

= (60 + 119) mod 256

= 179 mod 256

j = 179

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

swap = (S [5] , S [179])

Array S = [115, 213, 71, 191, 55, 179, 6, ..., 59, 4, 56, ..., 70, 2, 72, ..., 119,
0, 116, ... 170, 171, 172, 173, 5, 175, ..., 255]

Iterasi ketujuh ——> i = 6

j = 179

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

= (179 + S [6] + k [6 mod 8]) mod 256

= (179 + 6 + k [6]) mod 256

= (180 + "a") mod 256 -> ascii a = 97 (desimal)

= (180 + 97) mod 256

= 277 mod 256

j = 21

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

Swap = (S [6] , S [21])

Array S = [115, 213, 71, 191, 55, 179, 21, 7, ..., 20, 6, 22, ..., 59, 4, 56,
..., 70, 2, 72, ..., 119, 0, 116, ..., 172, 173, 5, 175, ..., 255]

Iteran kedelapan ——> i = 7

j = 21

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

j = [21 + 7 + k [7 mod (8)]) mod 256

j = (28 + ~~49~~ k7) mod 256

j = (28 + 49) mod 256

j = 77 mod 256

j = 77 mod 256

j = 77

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

Swap: (S[7], S[77])

array S= [115, 213, 71, 191, 55, 174, 21, 77, 8, 9, ..., 20, 6, 22, ..., 54, 9,
56, ..., 70, 2, 72, 73, 74, 75, 76, 7, 78, ..., 114, 0, 116, ..., 173,
5, 175, ..., 190, 3, 192, ..., 212, 1, 214, ..., 253, 254, 255]

2. PGRA

Array S = [115, 213, 71, 191, 55, 179, 21, 177, 8, 9, 10, ..., 20, 6, 22, ..., 54, 9, 56, ..., 70, 2, 72, ..., 76, 7, 78, ..., 119, 0, 116, ..., 173, 5, 175, ..., 190, 3, 192, ..., 212, 1, 219, ..., 254, 255]

Plainteks / P = 2094

Iteras pertama → i = 0

j = 0

for index = 0 to len(P) - 1
= 0 to (4) - 1 = 0 to (3)

i = (i + 1) mod 256

i = (0 + 1) mod 256

i = 1

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

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

j = (0 + 213) mod 256 = 213 mod 256

j = 213

Swap (S[i], S[j]) = (S[1], S[213])

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

t = 1 + 213 mod 256 = 219 mod 256

t = 219

u = S[219]

c = u ⊕ P[0]

= 219 ⊕ 2

= 11010110

00110010 ⊕

11100100 = 228 = ä

iterasi kedua ⟶ i = 1

j = 213

for index = 0 to (3)

i = (i+1) mod 256

i = (1+1) mod 256

i = 2

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

j = (213 + S[2]) mod 256

j = (213 + 71) mod 256 = 284 mod 256

j = ~~(215+71)~~ 28

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

t = (S[2] + S[28] mod 256

t = (28+71) mod 256 = 99 mod 256

t = 99

u = S[99]

C = u ⊕ P P[1]

= 99 ⊕ 0

= ~~0 0~~ 1 1 0 0 0 1 1

~~0 0~~ 1 0 0 0 0 0 ⊕

~~0 1 0 1 0 0~~

0 1 1 0 0 0 1 1

0 0 1 1 0 0 0 0 ⊕

0 1 0 1 0 0 1 1 = 83 = S (kapital)

iterasi ketiga ⟶ i = 2

j = 28

for indeks = 0 to (3)

i = (i+1) mod 256

i = (2+1) mod 256

i = 3

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

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

j = (28 + 191) mod 256 = 219 mod 256

j = 219

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

t = (S[3] + S[219]) mod 256

t = (219 + 191) mod 256 = 410 mod 256

t = 154

u = S[154]

C = u ⊕ p[2]
= 154 ⊕ 9
= 1001 1010
0011 1001 ⊕
1010 0011 = 163 = £

iterasi kempat → i = 3
j = 219
for indeks = 0 to (3)
i = (i+1) mod 256
i = (3+1) mod 256
i = 4
j = (j + s[i]) mod 256
j = (219 + S[3]) mod 256
j = (219 + 55) mod 256 = 274 mod 256
j = 18
swap = (S[i], S[j])
= (S[4], S[18])
t = (S[4] + S[18]) mod 256
t = (18 + 55) mod 256
= 73
u = S[73]
c = u ⊕ p[3]
= 73 ⊕ 4
= 0100 1001
0011 0100
0111 1101 = 125 = (}
~~= 125~~