* Algoritma : Key Scheduling Algorithm (KSA)

Kunci : "saputra", len (k) = 8

Array s : [0,1,2,3,4,5,6,7,8, ... 100,101,102,103, ... 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]) % 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[i] + k[i % 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, 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, 119, 0, 116, ..., 210, 211, 212, 1, 219, ..., 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

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

= (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, 213, 119, 0, 116, ..., 189, 190, 3, 192, ..., 210, 211, 212, 1, 219, ..., 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 + "t") % 256 ⇒ desimal "t" = 116

= (195 + 116) % 256

= 311 % 256

j = 55

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

swap (S[4], S[55])

Array S = [115, 213, 71, 191, 55, 5, 6, 7, 8, ..., 53, 54, 4, 56, 57, ..., 69, 70, 2, 72, 73, ..., 250, 251, 252, 253, 254, 255]

* Iterasi keenam → i = 5

j = 55

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

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

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

= (60 + "r") % 256 ⇒ desimal "r" = 114

= (60 + 114) % 256

= 174 % 256

= 174

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

swap (S[5], S[55])

Array S = [115, 213, 71, 191, 55, 174, 6, 7, 8, ..., 53, 54, 4, 56, 57, ..., 69, 70, 2, 72, 73, ..., 113, 114, 0, 116, 117, ..., 172, 173, 5, 175, 176, ..., 189, 190, 3, 192, 193, ..., 211, 212, 1, 214, 215, ..., 250, 251, 252, 253, 254, 255]

* Iterasi ketujuh → i = 6

J = 179

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

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

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

= (180 + "a") % 256 ≈ desimal "a" = 97

= (180 + 97) % 256

= 277 % 256

J = 21

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

swap (S[6], S[179])

Array S = [115, 213, 71, 191, 55, 179, 21, 7, 8, ..., 19, 20, 6, 22, 23, ... 53, 54, 4, 56, 57, ..., 69, 70, 2, 73, ... 113, 119, 0, 116, 117, ..., 172, 173, 5, 175, 176, ..., 189, 190, 3, 192, 193, ..., 211, 212, 1, 214, 215, ... 250, 251, 252, 253, 254, 255]

* Iterasi kedelapan → i = 7

J = 21

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

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

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

= (28 + "1") % 256 ≈ desimal "1" = 49

= (28 + 49) % 256

= 77 % 256

J = 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, ... 53, 54, 4, 56, 57, ..., 69, 70, 2, 72, 73, 79, 75, 76, 7, 78, ..., 113, 119, 0, 116, 117, ..., 172, 173, 5, 175, 176, ..., 189, 190, 3, 192, 193, ..., 211, 212, 1, 214, 215, ..., 250, 251, 252, 253, 254, 255]

Nama : Ilan Herlambang
NIM : E1E120010
Kelas : Genap
Mata Kuliah : Kriptografi

Pseudo Random Generation Algorithm (PRGA)

Plaintext = 20089

- Iterasi pertama

i = 0 j = 0

for idx = 0 to length (P) - 1 do
= 0 to len (S) - 1 do
= 0 to 9 do

i = (i + 1) mod 256
i = (0 + 1) mod 256
i = 1
j = (j + S[i]) mod 256
j = (0 + 213] mod 256 // nilai i diambil dari Array sebelumnya di KSA
j = 213 mod 256
j = 213

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

t = [S[1], + S[j]] mod 256
u = S[t]
= (1E + 213) mod 256
= 219 mod 256
t = 219
= u ⊕ S[219]
C = u ⊕ P[0]
= 219 ⊕ 2
⇒ Binary ⇒ 219 ⇒ 11 01 0110
2 ⇒ 00 11 0010 ⊕ XOR
11 10 01 00 → 228 ⇒ a

- Iterasi ke-2

i = 1, j = 213

For index = 0 to 9

i = (i+1) mod 256

i = (1+1) mod 256

= 2 mod 256

= 2

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

= (213 + S[2]) mod 256

= (213 + 71) mod 256

= 284 mod 256

j = 28

t = (S[i], S[j]) : (S[2], S[28])

= (S[2], + S[28]) mod 256

= (71 + 28) mod 256

= 99

C = U ⊕ P[1]

= 99 ⊕ 0

⇒ 01100011

00110000 ⊕

01010011 → chr ⇒ S (kapital)

- Iterasi ke-3

i = 2 j = 28

For idx = 0 to 90 do

i = (2+1) mod 256

i = 3 mod 256

i = 3

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

= (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

= (219 + 191) mod 256

= 410 mod 256

= 154

U = S [154]

C : U ⊕ P[2]

= 154 ⊕ 0

= 10011010
  00110000
  10101010 Dec = 170 ascii = ª

- Iterasi ke-4

i = 3 j = 215

For idx = 0 to 4 do

i = (3 + 1) mod 256

= 4

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

= (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

= (18 + 55) mod 256

= 73

u = S[73]

c = u ⊕ P[5]

= 73 ⊕ 1

Binary = 01001001
  00111000 ⊕
  01110001 Dec = 113 ascii = q

- Iterasi ke-5

i = 4 j = 18

For idx = 0 to 4 do

i = (4 + 1) mod 256

= 5

j = (18 + 174) mod 256

= 192 mod 256

j = 192

swap : S[i] . S[j] = S[S] . S[192]

t = (192 + 179) mod 256

= (366) mod 256

t = 110

u = S[110]

c = u ⊕ P[9] ⇒ = 110 ⊕ 0

= 01101110

00111000 ⊕

01011010

Dec = 90

ascii = Z (kapital)