* Algoritma : Key - Scheduling Algoritma (KSA)

kunci : Saptika I , len (k) = 8

Array S = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ..., 100, 101, 102, 103, ...
253, 254, 255]

* Iterasi Pertama → i = 0

J = 0

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

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

= [k[0]] % 256

= [S] % 256 → nilai desimal

= 115 % 256

J = 115

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

Swap (S[0],

Array S: [115, 1, 2, 3, 4, 5, 6, 7, ..., 110, 111, 112, 113, 114, 0, 116, ..., 117
210, 211, 212, 213, ..., 250, 251, 252, 253, 254, 255].

199, 200, 201, 202, 203, 204, 205, ..., 250, 251, 252, 253, 254, 25

* iterasi kedua → i = 1

J = 115

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

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

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

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

= (116 + "a") % 256 → decimal dari "a" = 97

= (115 + 97) % 256

= 213 % 256

J = 213

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

Swap (S[1], S[213])

Array S = [45, 213, 2, 3, 9, 5, 6, 7, ..., 112, 113, 119, 0, 116, ..., 210, 211, 212, 1, 219, ..., 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 + 77^27] % 256 → desimal dari 77^27 = 112

= [215 + 112] % 256

= 327 % 256

j = 71

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

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

Array S = [45, 213, 71, 3, 9, 5, 6, 7, ..., 69, 70, 2, 72, ..., 112, 113, 119, 0, 116, ..., 210, 211, 219, ..., 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

= [79 + 77^77] % 256 → desimal dari 77^77 = 117

= [79 + 117] % 256

= 191 % 256

j = 191

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

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

Array S = [115, 213, 71, 191, 9, 5, 6, 7, ..., 69, 70, 2, 72, ..., 112, 113, 119, 0, 116, ..., 210, 211, 212, 1, 219, ..., 250, 251, 252, 253, 254, 255].

* iterasi kelima → i = 4

T = 191

T = (T + 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[T])

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, ..., 113, 114, 0, 116, 117, ..., 189, 190, 3, 192, ..., 211, 212, 1, 214, ..., 250, 251, 252, 253, 254, 255}

* iterasi keenam → i = 5

T = 55

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

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

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

= [illegible]

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

= (60 + 114) % 256

= 174 % 256

j = 174

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

swap (S[5], S[174])

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 kebutuh → i = 6

t = 179

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

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

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

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

= (180 + 'a') % 256 → desimal 'a' = 97

= (180 + 97) % 256

= ~~0~~ 277 % 256

t = 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, 59, 9, 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 kelapan → i = 7

j = 21

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

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

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

= (28 + 'l') % 256 → desimal 'l' = 89

= (28 + 89) % 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, ...,
53, 59, 9, 89, 57, ..., 69, 70, 2, 72, 73, 179, 75, 76, 72, ...,
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]

Date:

* Algoritma : Pseudo-random Generation Algorithm

Array S = [115, 213, 70, 191, 55, 179, 21, 77, 8, ..., 19, 20, 6, 22, 23, ...,
53, 59, 9, 56, 57, ..., 69, 70, 2, 72, 73, 79, 75, 76, 7, 78, ...,
113, 114, 0, 116, 117, ..., 172, 173, 5, 175, 176, ..., 189, 190, 3, 192,
193, ..., 211, 212, 1, 212, 213, ..., 250, 251, 252, 253, 254, 255]

Plaintext = "20 81"

* iterasi pertama ↔ idx = 0

i = 0

j = 0

→ i = (i+1) % 256
= (0+1) % 256
= 1 % 256
= 1

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

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

Array S = {115, 1, 71, 191, 55, 179, 21, 77, 8, ..., 19, 20, 6, 22, 23, ...,
53, 59, 9, 56, 57, ..., 69, 70, 2, 72, 73, 79, 75, 76, 7, 78, ...,
113, 114, 0, 116, 117, ..., 172, 173, 5, 175, 176, ..., 189, 190, 3, 192,
193, ..., 212, 213, 212, ..., 250, 251, 252, 253, 254, 255}

* t = (S[i] + S[j]) % 256
= (S[1] + S[213]) % 256
= (1 + 213) % 256
= 214.

→ o = S[68]

= S[214] = 214 → = 11010110

→ C = U ⊕ P[idx]

= U ⊕ P[0]

= U ⊕ "z" → biner "z" = 1100010

= 11010110

01100010 ⊕

C = "ä" didefinisikan menjadi 228

* iterasi kedua → idx = 1

i = 1

j = 213

→ i = [i+1] % 256

= [1+1] % 256

= 2

→ j = [j + S[i]] % 256

= [213 + S[2]] % 256

= [213 + 71] % 256

= 284 % 256

= 28

swap[S[i], S[j]]

swap[S[2], S[28]]

∴ Array S = [115, 28, 191, 55, 179, 21, 77, 8, ..., 69, 20, 6, 22, 23, ...,
26, 27, 71, 29, 30, ..., 53, 54, 9, 56, 57, ..., 69, 70, 72,
73, 74, 75, 76, 7, 28, ..., 113, 114, 0, 116, 117, ..., 172,
173, 5, 175, 176, ..., 189, 190, 3, 192, 193, ..., 212,
214, 215, ..., 250, 251, 252, 253, 254, 255].

→ t = [S[i] + S[j]] % 256

= [S[2] + S[28]] % 256

= [28 + 71] % 256

= 99 % 256

= 99

→ o = S[t]

= S[99]

= 99 → biner 99 = 1100011

C = U ⊕ P[idx]

= U ⊕ P[1]

= 10 "0" → biner "0" = 110000

= 1100011

110000 ⊕

1010011

C = "S", desimal = 83

* Iterasi ketiga → idx = 2

i = 2 , j = 28

→ i = (i+1) % 256
= (2+1) % 256
= (2+1) % 256
= 3

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

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

swap(S[3], S[219])

Array S = {115, 1, 28, 219, 179, 21, 77, 8, ..., 16, 17, 55, 19, 20, 9, 22, 23, 24, 25, 27, 71, 29, 30, ..., 53, 54, 9, 56, 57, 69, 70, 2, 73, 79, 75, 76, 7, 78, 79, ..., 113, 114, 0, 116, 117, ..., 172, 173, 5, 175, 176, ..., 189, 190, 191, 192, 193, ..., 211, 212, 213, 215, 216, 217, 218, 219, 252, 253, ..., 220, 191, 222, 255}.

→ t = (S[i] + S[j]) % 256
= (S[3] + S[219]) % 256
= (219 + 191) % 256
= 410 % 256
= 154

→ K = S[t]
= S[154]
= 154 → biner 154 = 10011010

→ C = U ⊕ P[idx]
= U ⊕ P[2]
= U ⊕ 77_{16} → biner 77_{16} = 111000

10011010
00111000 ⊕
10100101

~~C = 77_{16}, desimal = 119~~

C = $A5_{16}$, desimal = 165

Date:

* iterasi: keempat → idx = 3

i = 3, J = 219

→ i = [i + 1] % 256
= [3 + 1] % 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, 179, 21, 77, 8, ..., 16, 17, 55, 19, 20, 6, 72, 23, 24, 25, 26, 27, 71, 29, 30, ..., 53, 54, 1, 57, 69, 70, 2, 73, 74, 75, 76, 7, 78, 79, ..., 113, 114, 0, 116, 117, ..., 172, ..., 173, 5, 176, ..., 189, 190, 3, 192, 193, ... 212, 213, 214, 215, 216, 217, 218, 191, 220, ..., 253, 254, 255].

→ t = [S[i] + [J]] % 256
= [S[4] + S[18]] % 256
= [18 + 55] % 256
= 73

→ v = S[t]
= S[73]
= 73 → biner 73 = 01001001

→ C = 0 ⊕ P[idx]
= 0 ⊕ P[3]
= 0 ⊕ "1" → biner "1" = 110001
= 01001001
00110001 ⊕
01111000

C = "x", desimal = 120 //

Dipindai dengan CamScanner