CSA5122-CRYPTOGRAPHY FOR NETWORK AND SECURITY

LAB PROGRAMS EXECUTION

15.RC5 ALGORITHM

import struct

# Constants

w, r, b = 32, 12, 16

mod = 2 \*\* w

P, Q = 0xB7E15163, 0x9E3779B9

def key\_expand(key):

L = [int.from\_bytes(key[i:i+4], 'little') for i in range(0, b, 4)]

S = [(P + i \* Q) % mod for i in range(2 \* (r + 1))]

A = B = i = j = 0

for \_ in range(3 \* max(len(S), len(L))):

A = S[i] = (S[i] + A + B) << 3 | (S[i] + A + B) >> (w - 3)

A %= mod

B = L[j] = (L[j] + A + B) << (A + B) % w | (L[j] + A + B) >> (w - (A + B) % w)

B %= mod

i, j = (i + 1) % len(S), (j + 1) % len(L)

return S

def encrypt(pt, S):

A, B = struct.unpack('<2I', pt)

A = (A + S[0]) % mod

B = (B + S[1]) % mod

for i in range(1, r + 1):

A = ((A ^ B) << (B % w) | (A ^ B) >> (w - B % w)) % mod

A = (A + S[2 \* i]) % mod

B = ((B ^ A) << (A % w) | (B ^ A) >> (w - A % w)) % mod

B = (B + S[2 \* i + 1]) % mod

return struct.pack('<2I', A, B)

def decrypt(ct, S):

A, B = struct.unpack('<2I', ct)

for i in range(r, 0, -1):

B = (B - S[2 \* i + 1]) % mod

B = ((B >> (A % w)) | (B << (w - A % w))) % mod ^ A

A = (A - S[2 \* i]) % mod

A = ((A >> (B % w)) | (A << (w - B % w))) % mod ^ B

return struct.pack('<2I', (A - S[0]) % mod, (B - S[1]) % mod)

# Test

key = b'RC5Demo16ByteKy'

msg = b'8byteMSG'

S = key\_expand(key)

ct = encrypt(msg, S)

pt = decrypt(ct, S)

print("Encrypted:", ct.hex())

print("Decrypted:",pt)

