**COEN 350: Network Security**

**ASSIGNMENT 2**

**Shweta Kharat**

**Date: 07/06/2019**

**RC4 Stream Cipher**

**Code (Java):**

**public** **class** RC4 {

**public** **static** **void** main(String[] args) {

// CASE 1: l=256, K[i]=0 for 0<=i<=255

// Key scheduling algorithm (KSA)

**int** S[] = **new** **int**[256];

**int** K[] = **new** **int**[256];

**int** l = 256;

// Initialization

**for**(**int** i=0; i<256; i++){

S[i] = i;

K[i] = 0;

}

// Initial permutation

**int** j = 0;

**int** temp = 0;

**for**(**int** i=0; i<256; i++){

j = (j + S[i] + K[i%l] ) % 256;

temp = S[i];

S[i] = S[j];

S[j] = temp;

}

// Subkey Generation Algorithm (SGA)

// Initialization

**int** i=0; j=0; temp=0;

// Permutation and generation loop

**for**(**int** u=0; u<256; u++){

i = (i+1) % 256;

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

temp = S[i];

S[i] = S[j];

S[j] = temp;

K[u] = S[(S[i] + S[j]) % 256];

}

// Print first 20 bytes of key generated

System.***out***.println("Case 1: First 20 keys are");

**for**(**int** k=0; k<20; k++){

System.***out***.print(K[k] + " ");

}

// CASE 2: l=5, K[0] = 15;K[1] = 202;K[2] = 33;K[3] = 6;K[4] = 8;

// Key scheduling algorithm (KSA)

**int** S2[] = **new** **int**[256];

**int** K2[] = **new** **int**[256];

**int** l2 = 5;

K2[0] = 15;

K2[1] = 202;

K2[2] = 33;

K2[3] = 6;

K2[4] = 8;

// Initialization

**for**(**int** i2=0; i2<256; i2++){

S2[i2] = i2;

}

// Initial permutation

**int** j2 = 0;

**int** temp2 = 0;

**for**(**int** i2=0; i2<256; i2++){

j2 = (j2 + S2[i2] + K2[i2%l2] ) % 256;

temp2 = S2[i2];

S2[i2] = S2[j2];

S2[j2] = temp2;

}

// Subkey Generation Algorithm (SGA)

// Initialization

**int** i2=0; j2=0; temp2=0;

// Permutation and generation loop

**for**(**int** u2=0; u2<256; u2++){

i2 = (i2+1) % 256;

j2 = (j2 + S2[i2]) % 256;

temp2 = S2[i2];

S2[i2] = S2[j2];

S2[j2] = temp2;

K2[u2] = S2[(S2[i2] + S2[j2]) % 256];

}

// Print first 20 bytes of key generated

System.***out***.println("\nCase 2: First 20 keys are");

**for**(**int** k2=0; k2<20; k2++){

System.***out***.print(K2[k2] + " ");

}

}

}

**Output:**

Case 1: Let K (i) = 0; for 0<= i<= 255: In this case l = 256: This specifies the initial key

vector. Output first 20 bytes of the generated key stream in decimal notation.

Case 2: Let K (0) = 15; K (1) = 202; K (2) = 33; K (3) = 6; and K (4) = 8: In this case

l = 5: Output first 20 bytes of the generated key stream in decimal notation.

Terminal output:

