RC4

#include <stdio.h>

#include <string.h>

#define MAX\_TEXT\_LENGTH 1024 // Define a maximum size for the plaintext/ciphertext

#define SWAP(a, b) { int temp = a; a = b; b = temp; }

void rc4\_init(unsigned char \*key, int key\_len, unsigned char \*S) {

int i, j = 0;

// Initialize the permutation array S

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

S[i] = i;

}

// Key-scheduling algorithm (KSA)

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

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

SWAP(S[i], S[j]);

}

}

void rc4\_generate(unsigned char \*S, unsigned char \*input, unsigned char \*output, int data\_len) {

int i = 0, j = 0, t;

// Pseudo-random generation algorithm (PRGA)

for (int k = 0; k < data\_len; k++) {

i = (i + 1) % 256;

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

SWAP(S[i], S[j]);

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

output[k] = input[k] ^ S[t]; // XOR input with generated keystream

}

}

int main() {

unsigned char key[256]; // Buffer to hold the key

unsigned char plaintext[MAX\_TEXT\_LENGTH]; // Buffer to hold the plaintext

int key\_len;

// Get the key from the user

printf("Enter the key: ");

fgets((char \*)key, sizeof(key), stdin);

key\_len = strlen((char \*)key) - 1; // Exclude newline character

// Get the plaintext message from the user

printf("Enter the plaintext: ");

fgets((char \*)plaintext, sizeof(plaintext), stdin);

int data\_len = strlen((char \*)plaintext) - 1; // Exclude newline character

unsigned char S[256]; // State array

unsigned char ciphertext[MAX\_TEXT\_LENGTH];

unsigned char decrypted[MAX\_TEXT\_LENGTH];

// Initialize the RC4 cipher with the key

rc4\_init(key, key\_len, S);

// Encrypt the plaintext

rc4\_generate(S, plaintext, ciphertext, data\_len);

printf("Ciphertext: ");

for (int i = 0; i < data\_len; i++) {

printf("%02X ", ciphertext[i]); // Print each byte as hexadecimal

}

printf("\n");

// Reinitialize S for decryption (RC4 encryption and decryption are symmetric)

rc4\_init(key, key\_len, S);

// Decrypt the ciphertext

rc4\_generate(S, ciphertext, decrypted, data\_len);

printf("Decrypted text: ");

for (int i = 0; i < data\_len; i++) {

printf("%c", decrypted[i]);

}

printf("\n");

return 0;

}