Reg No= ch.sc.p2cse24006

Name = Vishal Vinod Kumar

Dept = CSE

Course= M.Tech

LFSR Implementation Assignment

Aim- To write a code to Implement LFSR and encrypt the given message.

Parameters:

Input Used = “HELLO”

Key Used= “10111011”

Recurrence Relation used = S[i+4] = S[i]+S[i+1]

Code for LFSR:

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

char intToChar(int n) {

    // Convert int to char by adding '0' to the int

    return n + '0';

}

int charToInt(char c) {

    // Convert char to int by subtracting '0' from the char

    return c - '0';

}

void stringToBinary(char \*str, char \*binary) {

    // Cpoy the binary representation of each character in the string to the binary string

    while (\*str) {

        for (int i = 7; i >= 0; i--) {

            \*binary++ = (\*str & (1 << i)) ? '1' : '0';

        }

        str++;

    }

    \*binary = '\0';

}

int areStringsSameLength(char \*str1, char \*str2) {

    //Check if the strings are of the same length

    int len1 = 0, len2 = 0;

    while (\*str1++) len1++;

    while (\*str2++) len2++;

    return len1 == len2;

}

void lfsr(char \*key,char \*input\_binary , char \*extended\_key) {

    // Extend the key to the length of the input binary

    int i = 0;

    for (i = 0; i < strlen(input\_binary); i++) {

        extended\_key[i+8] = intToChar(charToInt(extended\_key[i])^ charToInt(extended\_key[i+1]));

    }

}

void encrypt(char \*input\_binary, char \*key, char \*CipherBinary) {

    // Encrypt the input binary using the key using one Time Pad

    int i = 0;

    for (i = 0; i < strlen(input\_binary); i++) {

        CipherBinary[i] = intToChar(charToInt(input\_binary[i]) ^ charToInt(key[i]));

    }

    CipherBinary[i] = '\0';

}

void decrypt(char \*CipherBinary, char \*key, char \*DecryptedBinary) {

    // Decrypt the Cipher Binary using the key using one Time Pad

    int i = 0;

    for (i = 0; i < strlen(CipherBinary); i++) {

        DecryptedBinary[i] = intToChar(charToInt(CipherBinary[i]) ^ charToInt(key[i]));

    }

    DecryptedBinary[i] = '\0';

}

void binaryToString(char \*binary, char \*str) {

    // Convert the binary string to a string

    while (\*binary) {

        char byte = 0;

        for (int i = 0; i < 8; i++) {

            byte = (byte << 1) | (\*binary++ - '0');

        }

        \*str++ = byte;

    }

    \*str = '\0';

}

int main() {

    char input[100]="HELLO";

    char key[100] ="10111011";

    char input\_binary[100],extended\_key[100],CipherBinary[100],CipherText[100],decryptText[100];

     int i= 0;

     while (key[i] != '\0') {

            extended\_key[i] = key[i];

            i++;

        }

        extended\_key[i] = '\0';

    stringToBinary(input, input\_binary);

    printf("String: %s\n", input);

    printf("Binary String: %s\n", input\_binary);

    if (!areStringsSameLength(key, input\_binary)) {

        lfsr(key,input\_binary,extended\_key);

    }

        printf("Key: %s\n", key);

        printf("Externded Key: %s\n", extended\_key);

    encrypt(input\_binary, extended\_key, CipherBinary);

    printf("Cipher Binary Text: %s\n", CipherBinary);

    decrypt(CipherBinary, extended\_key, decryptText);

    printf("Cipher Decrypt Text: %s\n", decryptText);

    return 0;

}

OUTPUT

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated