**Description for: CS 492 Homework 2**

Symmetric Cryptography

**You may work in groups of 2**

Complete the problems below and submit this word document with the solution to the questions here. Also submit the source code **you** created to solve problem 2

Name(s): Matthew Krieger  
  
**Problem 1 Stream encryption (30 pts):**

You have a 4 bit LFSR with tap sequence [3,2,0] your initial key is 1101 encrypt the message “1001” and enter it below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Bit 0 | Bit 1 | Bit 2 | Bit 3 | Key stream | Plain Text | Encrypted message |
| 1 | 1 | 0 | 1 |  |  |  |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 |

**0010**

**Problem 2 Block encryption (40 pts):**

**YOU** implement the Tiny Encryption Algorithm (TEA) in either Java or C++

**Part 1:** Use your TEA algorithm to encrypt the 64-bit plaintext block:

0x0123456789ABCDEF

Using 128 bit key:

0xA56BABCDEF00F000FFFFFFFFABCDEF01

Enter your cipher text here:

**0xe1881bde581234e6**

**Part 2:** Implement decryption and verify that you obtain the original plain text

|  |
| --- |
| import java.util.Scanner;  public class TEAEncryption {  public static Long encrypt(Long[] lKey,Long plainText,Long delta) {  // declarations  Long total = 0L;  int iterations = 32;  // Shift right zero fill operator. plainText's value is moved right by 32  Long left =  plainText>>>iterations;  Long cipherlong = 0L;  Long bitMask = 0x00000000FFFFFFFFL;  // bitwise AND used on plainText and bitMask, assigned to right  Long right = plainText & bitMask;        // Pseudocode from page 71  for (int i=0; i<iterations; i++) {  // Pseudocode from page 71  //add delta to total  total = total + delta;  // bitwise AND used on total(delta) and bitMask  total = total & bitMask;  // Pseudocode from page 71  left = left + (((right<<4)+lKey[0]) ^ (right+total) ^ ((right>>>5)+lKey[1]));  // biwise AND used on left and bitMask, assigned to left  left = left & bitMask;  // Pseudocode from page 71  right = right + (((left<<4)+lKey[2]) ^ (left+total) ^ ((left>>>5)+lKey[3]));  // bitwise AND used on right and bitmask, assigned to right  right = right & bitMask;  // bitwise AND used on total and bitMask, assigned to total  total = total & bitMask;  }  // bitwise AND used on right and bitMask, assigned to right  right = right & bitMask;  // bitwise AND used on left and bitmask then ==> Binary Left Shift Operator is used  left = (left & bitMask) << iterations;  // Pseudocode from page 71  // bitwise OR used on left & right and returned to main  return cipherlong = left | right;    }  public static Long decrypt(Long[] lKey,Long lEncrypt, Long delta) {  // declarations  // Pseudocode from page 72(Stamp)  Long total  = delta << 5;  int iterations = 32;  Long bitMask = 0x00000000FFFFFFFFL;      Long left =  lEncrypt >>> iterations;  Long right = lEncrypt & bitMask;  // Pseudocode from page 72(Stamp)  for (int i=0; i<iterations; i++) {    //bitwise AND used on total and bitMask / assigned to total  total = total & bitMask;  // Pseudocode from page 72(Stamp)  right = right - (((left<<4)+lKey[2]) ^ (left+total) ^ ((left>>>5)+lKey[3]));  //bitwise AND used on right and bitMask / assigned to right  right = right & bitMask;  // Pseudocode from page 72(Stamp)  left = left - (((right<<4)+lKey[0]) ^ (right+total) ^ ((right>>>5)+lKey[1]));  //bitwise AND used on left and bitMask / assigned to left  left = left & bitMask;  // Pseudocode from page 72(Stamp)  total = total - delta;  // bitwise AND used on total and bitmask  total = total & bitMask;  }  // bitwise AND used on right and bitMask  right = right & bitMask;  // bitwise AND used on left and bitMask  left = left & bitMask;  // Binary left shift by 32 bits  left = left << iterations;  // bitwise OR used on left and right  Long lDecrypt = left | right;  Long dMask = 0x7FFFFFFFFFFFFFFFL;  // bitwise AND used on lDecrypt and dMask  return lDecrypt = lDecrypt & dMask;    }  public static void main(String args[]) {  //declarations  Scanner scanner = new Scanner( System.in );  // asks user to input key  System.out.print("Enter A Key: ");  // assign key to strKey  String strKey = scanner.nextLine();  scanner.close();  //assign plaintext  Long plainText = 0x0123456789ABCDEFL;          //assign delta  Long delta = 0x9e3779b9L;  Long[] lKey = new Long[4];  String[] sKey = new String[4];  sKey[0] = strKey.substring(2, 10);  sKey[1] = strKey.substring(10, 18);  sKey[2] = strKey.substring(18, 26);  sKey[3] = strKey.substring(26, 34);    for (int i = 0; i< lKey.length;i++){  lKey[i] = Long.parseLong(sKey[i],16);  }    //call encrypt function and assign value to lEncrypt  Long lEncrypt = encrypt(lKey, plainText, delta);  // call decrypt function and assign value to lDecrrypt  Long lDecrypt = decrypt(lKey,lEncrypt, delta);  // format for hex  String hexa = "0x";  // print plainText  System.out.println("Plaintext: " + hexa + Long.toHexString(plainText));  //print lEncrypt  System.out.println("Cyphertext: "+ hexa + Long.toHexString(lEncrypt));  //print lDecrypt  System.out.println("Decrypted text: "+ hexa + Long.toHexString(lDecrypt));    }  } |

**Problem 3 Chain block encryption (30 pts)**

Given the following message and a block size of 2 bits encode it using ECB, CBC, and CTR (make sure to indicate which is which)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Use the following substitution cipher | | | | Plain | 00 | 01 | 10 | 11 |
|  |  |  |  | Encoded | 10 | 00 | 11 | 01 |
| IV | 0 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| Message | 10 | 00 | 11 |  |  |  |  |  |

ECB:

Co = E(Po, K)

Po= D(Co, K)

M = 10 00 11

**C = 11 10 01**

CTR

Co = Po, ⊕ E(IV, K)

C1 = P1 ⊕ E(IV+1, K)

M = 10 00 11

IV+ = 00 01 10

E(IV, K) 10 00 11

**C = 00 00 00**

CBC:

Co = E(IV⊕Po, K)

C1 = E(Co⊕P1, K)

M = 10 00 11

IV/Ci-1 = 00 11 01

⊕Pi = 10 11 10

**C = 11 01 11**