Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Лабораторна робота

із Криптографії №4

Побудова регістрів зсуву з лінійним зворотним зв’язком та дослідження їх властивостей

Виконали:

Студенти групи ФБ-74

Стурчак Максим та Харламова Катерина

Перевірено_____

Київ 2019

КРИПТОГРАФІЯ
КОМП'ЮТЕРНИЙ ПРАКТИКУМ №4
Побудова регістрів зсуву з лінійним зворотним зв'язком та дослідження їх властивостей

Мета роботи
Ознайомлення з принципами побудови регістрів зсуву з лінійним зворотним зв'язком; практичне освоєння їх програмної реалізації; дослідження властивостей лінійних рекурентних послідовностей та їх залежності від властивостей характеристичного полінома регістра.

Порядок виконання роботи
0. Уважно прочитати методичні вказівки до виконання комп'ютерного практикуму.
1. Вибрати свій варіант завдання згідно зі списком. Варіанти завдань містяться у файлі Crypto_CP4 LFSR_Var.
2. За даними характеристичними многочленами $p_1(x)$, $p_2(x)$ скласти лінійні рекурентні співвідношення для ЛРЗ, що задаються цими характеристичними многочленами.
3. Написати програми роботи кожного з ЛРЗ L_1 , L_2 .
4. За допомогою цих програм згенерувати імпульсні функції для кожного з ЛРЗ і підрахувати їх періоди.
5. За отриманими результатами зробити висновки щодо влавстивостей кожного з характеристичних многочленів $p_1(x)$, $p_2(x)$: многочлен примітивний над F_2 ; не примітивний, але може бути незвідним; звідний.
6. Для кожної з двох імпульсних функцій обчислити розподіл k-грам на періоді, $k \le n_i$, де n_i - степінь полінома $f_i(x)$, $i=1,2$ а також значення функції автокореляції $A(d)$ для $0 \le d \le 10$. За результатами зробити висновки.

Варіант 19:

P1(X)= X22+X18+X16+X13+X12+X11+X10+X5+X3+X2+1
P2(X)= X20 + X17 + X16 + X15 + X10 + X8 + X6 + X5 + X3 + X + 1

Довжини періодів:
L_1: 4194303 => $P_1(x)$ – примітивний
L_2: 15015 => $P_2(x)$ – не прімітивний, звідний

Кількість К-грам полінома P1:

2-грами:		3-грами		4-грами		5-грами	
00	524515	000	174421	0000	65802	00000	26361
01	524002	001	174592	0001	65388	00001	26136
10	524119	010	175104	0010	65823	00010	26069
11	524515	011	174592	0011	65267	00011	26195
		100	175104	0100	65154	00100	25923
		101	174592	0101	65462	00101	26478
		110	175104	0110	65424	00110	26337
		111	174592	0111	65850	00111	26031
				1000	65426	01000	26033
				1001	65421	01001	26215
				1010	65555	01010	26122
				1011	65411	01011	26284
				1100	65853	01100	25920
				1101	65841	01101	26173
				1110	65503	01110	26368
				1111	65395	01111	26399
						10000	26582
						10001	26350
						10010	26339
						10011	26358
						10100	10100
						10101	26109
						10110	26205
						10111	26151
						11000	26091
						11001	26136
						11010	26316
						11011	26161
						11100	26269
						11101	26337
						11110	26077
						11111	26132

Кількість К-грам полінома P2:

2-грами		3-грами		4-грами		5-грами	
00	1855	000	616	0000	232	00000	91
01	1918	001	616	0001	268	00001	82
10	1855	010	623	0010	224	00010	86
11	1879	011	630	0011	238	00011	99
		100	630	0100	224	00100	83
		101	623	0101	0101	00101	119
		110	644	0110	213	00110	78
		111	623	0111	255	00111	86
				1000	224	01000	119
				1001	263	01001	107
				1010	230	01010	91
				1011	260	01011	102
				1100	213	01100	78
				1101	247	01101	106
				1110	211	01110	98
				1111	227	01111	94
						10000	96
						10001	100
						10010	85
						10011	105
						10100	88
						10101	90
						10110	76
						10111	71
						11000	99
						11001	94
						11010	82
						11011	109
						11100	90
						11101	105
						11110	95
						11111	99

Значення автокореляції:

L_1:	L_2:
d = 1 : 2097152	d = 1 : 7518
d = 2 : 2097152	d = 2 : 7532
d = 3 : 2097152	d = 3 : 7504
d = 4 : 2097152	d = 4 : 7518
d = 5 : 2097152	d = 5 : 7476
d = 6 : 2097152	d = 6 : 7518
d = 7 : 2097152	d = 7 : 7518
d = 8 : 2097152	d = 8 : 7490
d = 9 : 2097152	d = 9 : 7490
d = 10 : 2097152	d = 10 : 7490

Висновок:

В лабораторній роботі було набуто навичок програмної реалізації з лінійними регістрами зсуву, дослідження властивостей характеристичного полінома регістра. Також було досліджено властивості лінійних рекурентних послідовностей

Код програми:

```
package com.company;

import java.io.FileOutputStream;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.io.IOException;
import java.util.List;

public class Main {

   public static void main(String[] args) throws IOException {
      String string = arr().toString().replaceAll("[^0-1]", "");
      fillMap(string);
      corelation(arr());

   }

   public static String smaller(int i) {
      StringBuilder s = new StringBuilder();
      s.append("(?<=\\G");
      for (int j = 0; j < i; j++)
         s.append(".");
      s.append(")");

      return s.toString();
   }

   static void fillMap(String string) throws IOException {
      String[] nGramms;
      HashMap<String, Float> map = new HashMap<>();
      for (int i = 1; i < 6; i++) {
         nGramms = string.split(smaller(i));
         for (String symb : nGramms) {
            if (!symb.isEmpty())
               if (map.containsKey(symb)) {
```

```
                map.put(symb, map.get(symb) + 1);
            } else {
                map.put(symb, (float) 1);
            }
        }

        String mString = map.toString();
        FileOutputStream fos = new FileOutputStream("file" + i + ".txt");
        System.out.println("Размер " + i + " грамм = " + map.size());
        fos.write(mString.getBytes());
        fos.flush();
        fos.close();
        map.clear();
    }
    //System.out.println(map);
}


public static ArrayList arr() {
   // int[] arr1 = {1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0};
     int[] arr1 = {1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0};
    int[] arr2 = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1};
    int[] a2 = Arrays.copyOf(arr2, arr2.length);
    int period = 0;
    ArrayList<Integer> arrResult = new ArrayList<>();
    int sum;
    do {
        period++;
        sum = sumOfMas(arr1, arr2);
        arrResult.add(arr2[0]);
        moveLeftTheArray(arr2, sum);
        //System.out.println(Arrays.toString(arr2));

    } while (!ifArr1EqArr2(a2, arr2));
    System.out.println("Период = " + period);

    return arrResult;
}
public static boolean ifArr1EqArr2(int[] arr1, int[] arr2) {
    return Arrays.equals(arr1, arr2);
}

public static int[] moveLeftTheArray(int arr[], int lastNumber) {
    int size = arr.length;
    for (int j = 0; j < size - 1; j++) {
        arr[j] = arr[j + 1];
    }
    arr[size - 1] = lastNumber;

    return arr;
```

```
    }

    public static int sumOfMas(int arr1[], int arr2[]) {
        int sum = 0;
        int multiply;
        for (int i = 0; i < arr1.length; i++) {
            multiply = arr1[i] * arr2[i];
            sum += multiply;
        }

        return sum % 2;
    }




    public static void corelation(ArrayList arrayList) {
        int mainSum = 0;
        int result;
        List<Integer> newArrList = new ArrayList<>();
        for (int i = 1; i < 11; i++) {
            for (int j = 0; j < arrayList.size(); j++) {
                result = (arrayList.get(j % arrayList.size()) == arrayList.get((j + i) % arrayList.size())) ? 0 : 1;
                newArrList.add(result);
                //j++;
            }

            for (int j = 0; j < newArrList.size(); j++) {
                mainSum += newArrList.get(j);
            }
            System.out.println("Сумма  " + i + " = " + mainSum);
            newArrList.clear();
            mainSum = 0;

        }
    }

}
```