Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Фізико-технічний інститут

Лабораторна робота
із Криптографії №4

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

Виконав:

студент 3 курсу ФТІ

групи ФБ-74,ФБ-72

Скуратов Илья, Демиденко Дарья

Перевірили:

Чорний О.

Савчук М. М.

Завадська Л. О.

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 $p1(x)$, $p2(x)$ скласти лінійні рекурентні співвідношення для ЛРЗ, що задаються цими характеристичними многочленами. 3. Написати програми роботи кожного з ЛРЗ $L1$, $L2$. 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 $p1(x)$, $p2(x)$: многочлен примітивний над $F2$; не примітивний, але може бути незвідним; звідний. 6. Для кожної з двох імпульсних функцій обчислити розподіл k-грам на періоді, $k \le ni$, де ni - степінь полінома $fi(x)$, i=1,2 а також значення функції автокореляції $A(d)$ для $0 \le d \le 10$. За результатами зробити висновки.

Варіант 15:

P1(X) = X23 + X20 + X17 + X16 + X14 + X12 + X10 + X9 + X8 + X7 + X3 + X + 1

P2(X) = X20 + X18 + X17 + X16 + X13 + X12 + X11 + X9 + X6 + X5 + 1

Довжини періодів:

T1 = 8388607

T2 = 349525

P1(x) – PRIMITIVE
P2(x) – NO PRIMITIVE, BUT REDUCIBLE

Розподіл К-грам полінома P1:

2-грами:
00 - 0.251783
01 - 0.248024
10 - 0.249655
11 - 0.250539

3-грами
000 - 0.124316
001 - 0.12609
010 - 0.124362
011 - 0.123801
100 - 0.125094
101 - 0.126273
110 - 0.124431
111 - 0.125632

4-грами
0000 - 0.0622559
0001 - 0.0629569
0010 - 0.0629283
0011 - 0.0624133
0100 - 0.0618554
0101 - 0.0618268
0110 - 0.063758
0111 - 0.0617838
1000 - 0.0639153
1001 - 0.0635005
1010 - 0.0615407
1011 - 0.0626851
1100 - 0.0609542
1101 - 0.0621415
1110 - 0.0636006
1111 - 0.061884

5-грами
00000 - 0.0313449
00001 - 0.0304867
00010 - 0.0303493
00011 - 0.0303493
00100 - 0.0328727
00101 - 0.0325122
00110 - 0.0303665
00111 - 0.0305725
01000 - 0.0313964
01001 - 0.0303665
01010 - 0.0301777
01011 - 0.0307785
01100 - 0.0319286
01101 - 0.0313964
01110 - 0.030212
01111 - 0.0302978
10000 - 0.0317226
10001 - 0.0313621
10010 - 0.0314994
10011 - 0.0310016
10100 - 0.0311905
10101 - 0.030521
10110 - 0.0324607
10111 - 0.0319801
11000 - 0.0323749
11001 - 0.0312076
11010 - 0.0316711
11011 - 0.0313793
11100 - 0.0318771
11101 - 0.0316539
11110 - 0.03083
11111 - 0.0318599

Розподіл К-грам полінома P2:

2-грами:
00 - 0.251783
01 - 0.248024
10 - 0.249655
11 - 0.250539

3-грами
000 - 0.124316
001 - 0.12609
010 - 0.124362
011 - 0.123801
100 - 0.125094
101 - 0.126273
110 - 0.124431
111 - 0.125632

4-грами
0000 - 0.0622559
0001 - 0.0629569
0010 - 0.0629283
0011 - 0.0624133
0100 - 0.0618554
0101 - 0.0618268
0110 - 0.063758
0111 - 0.0617838
1000 - 0.0639153
1001 - 0.0635005
1010 - 0.0615407
1011 - 0.0626851
1100 - 0.0609542
1101 - 0.0621415
1110 - 0.0636006
1111 - 0.061884

5-грами
00000 - 0.0313449
00001 - 0.0304867
00010 - 0.0303493
00011 - 0.0303493
00100 - 0.0328727
00101 - 0.0325122
00110 - 0.0303665
00111 - 0.0305725
01000 - 0.0313964
01001 - 0.0303665
01010 - 0.0301777
01011 - 0.0307785
01100 - 0.0319286
01101 - 0.0313964
01110 - 0.030212
01111 - 0.0302978
10000 - 0.0317226
10001 - 0.0313621
10010 - 0.0314994
10011 - 0.0310016
10100 - 0.0311905
10101 - 0.030521
10110 - 0.0324607
10111 - 0.0319801
11000 - 0.0323749
11001 - 0.0312076
11010 - 0.0316711
11011 - 0.0313793
11100 - 0.0318771
11101 - 0.0316539
11110 - 0.03083
11111 - 0.0318599

Значення автокореляції:

L1:

D = 1 : 4194304
D = 2 : 4194304
D = 3 : 4194304
D = 4 : 4194304
D = 5 : 4194304
D = 6 : 4194304
D = 7 : 4194304
D = 8 : 4194304
D = 9 : 4194304
D = 10 : 4194304

L2:

D = 1 : 174592
D = 2 : 174592
D = 3 : 174592
D = 4 : 174592
D = 5 : 174592
D = 6 : 174592
D = 7 : 175104
D = 8 : 174592
D = 9 : 174592
D = 10 : 174592

Висновок:

В даному комп'ютерному практикумі було набуто навичок роботи з лінійними регістрами зсуву, а саме: їх програмна реалізація, дослідження властивостей характеристичного полінома регістра. Окрім цього було досліджено властивості лінійних рекурентних послідовностей

Код:

```
#include "stdafx.h"
#include <iostream>
#include <vector>
#include <math.h>
#include <map>
#include<string>


using namespace std;


//P1(X) = X23 + X20 + X17 + X16 + X14 + X12 + X10 + X9 + X8 + X7 + X3 + X + 1
//P2(X) = X20 + X18 + X17 + X16 + X13 + X12 + X11 + X9 + X6 + X5 + 1



vector <int> v1{ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1 };
vector <int> v2{ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1 };




vector <int> orig1 = {1,1,0,1,0,0,0,1,1,1,1,0,1,0,1,0,1,1,0,0,1,0,0};
vector <int> orig2 = {1,0,0,0,0,1,1,0,0,1,0,1,1,1,0,0,1,1,1,1,0};

vector <int> buf1(90000000);
vector <int> buf2(90000000);


int sdvig(vector <int> koef_polynom, int kto, vector <int> skelet);
void kgram(vector <int> bufer, int shag);
int xop(vector <int> v, int shag, int per);
void polynom(int size, int per);

int period1 = 0;
int period2 = 0;


int main()
{
	period2 = sdvig(orig2,2,v2);
	period1 = sdvig(orig1, 1, v1);

	cout << "T1 = "<<period1<<endl;
	cout << "T2 = " << period2 << endl;

	cout << endl << "FIRST POLYNOM : " << endl<<endl;

	for (int i = 2; i < 6; i++)
	{
		cout << i << "-gramma :" << endl << endl;
		kgram(buf1, i);
		cout << endl;
```

```
	}

	cout << endl << "SECOND POLYNOM : " << endl << endl;

	for (int i = 2; i < 6; i++)
	{
		cout << i << "-gramma :" << endl << endl;
		kgram(buf1, i);
		cout << endl;
	}

	cout << endl << "FIRST POLYNOM : " << endl << endl;


	for (int i = 1; i < 11; i++)
	{
		xop(buf1, i,period1);

	}
	cout << endl << "SECOND POLYNOM : " << endl << endl;

	for (int i = 1; i < 11; i++)
	{
		xop(buf2, i,period2);
	}


	int size1 = orig1.size();
	int size2 = orig2.size();


	polynom(size1, period1);
	polynom(size2, period1);



	system ("pause");
	return 0;

}




int sdvig(vector <int> koef_polynom, int kto, vector <int> skelet)
{

	vector <int> copy ;
	int p = 0;
	int y = 0;
	int period = 0;

	copy = skelet;

	int sizee = koef_polynom.size();

	do
	{
		y = 0;
		p = 0;
		if (kto == 1) buf1[period] = skelet[0];
		if (kto == 2) buf2[period] = skelet[1];
		period++;
```

```
		for (int i = 0; i < sizee; i++)
		{
			y += skelet[i] * koef_polynom[i];

		}

		for (int i = 0; i < sizee-1; i++)
		{
			skelet[i] = skelet[i + 1];
		}


		skelet[sizee-1] = y % 2;

	}
	while (skelet!= copy);


	return period;
}


void kgram (vector <int> bufer, int shag)
{

	map <string, double> kgram;

	string newgram;

	int schetchik = 0;

	int sizeq = bufer.size();

	for (int i = 0; i < period2; i += 1 + shag) {

		if (i + shag <= sizeq)
		{


		for (int j = 0; j < shag; j++)
		{
			newgram+=(to_string(bufer[i + j]));
		}

		if (kgram.count(newgram)) {
			schetchik++;
			kgram.at(newgram)++;
		}
		else {
			schetchik++;
			kgram.emplace(newgram, 1);
		}
		newgram.clear();
	}


	}




	for (auto it = kgram.cbegin(); it != kgram.cend(); it++)
	{
```

```
                string tt;
                double itog;

                tt = it->first;
                itog = it->second / schetchik;

         cout << tt << " - " << " " << itog  << endl;

        }

}



int xop(vector <int> v, int shag, int per)
{
        int otvet = 0;
        for (int i = 0; i < per; i++)
        {
                otvet += (v[i] + v[(i + shag)%per]) % 2;

        }

        cout << "D = " << shag << " : " << otvet<<endl;



        return 0;
}


void polynom( int size, int per)
```