Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Фізико-технічний інститут

**ЛАБОРАТОРНА РОБОТА №2**

**з дисципліни**

**«Криптографія»**

**на тему: «**Криптоаналіз шифру Віженера **»**

Виконали:

студенти 3 курсу ФТІ

групи ФБ-72,ФБ-74

Демиденко Дар’я та Скуратов Ілля

Перевірили:

Чорний О.

Савчук М. М.

Завадська Л. О.

Мета роботи

Засвоєння методів частотного криптоаналізу. Здобуття навичок роботи та аналізу потокових шифрів гамування адитивного типу на прикладі шифру Віженера.

**Порядок виконання роботи**

0. Уважно прочитати методичні вказівки до виконання комп’ютерного практикуму.

1. Самостійно підібрати текст для шифрування (2-3 кб) та ключі довжини r = 2, 3, 4, 5, а також довжини 10-20 знаків. Зашифрувати обраний відкритий текст шифром Віженера з цими ключами.

2. Підрахувати індекси відповідності для відкритого тексту та всіх одержаних шифртекстів і порівняти їх значення.

3. Використовуючи наведені теоретичні відомості, розшифрувати наданий шифртекст (згідно свого номеру варіанта).

**Результати програми**

Index for open text = 0.0561828

index for k2 : 0.0417827

index for k3 : 0.0368259

index for k5 : 0.0364497

index for k9 : 0.0345875

index for k12 : 0.0348242

index for k15 : 0.0328668

index for k19 : 0.0326702

Index for 2 = 0.0353776

Index for 3 = 0.0338309

Index for 4 = 0.0353141

Index for 5 = 0.0337755

Index for 6 = 0.0354391

Index for 7 = 0.0424808

Index for 8 = 0.0354522

Index for 9 = 0.0338365

Index for 10 = 0.0352651

Index for 11 = 0.0337932

Index for 12 = 0.0353988

Index for 13 = 0.0337373

Index for 14 = 0.0534105

Index for 15 = 0.0336351

Index for 16 = 0.0357043

Index for 17 = 0.0338589

Index for 18 = 0.0354968

Index for 19 = 0.0340795

Index for 20 = 0.0352489

Index for 21 = 0.0423153

Index for 22 = 0.0351691

Index for 23 = 0.0337577

Index for 24 = 0.0355538

Index for 25 = 0.0339126

Index for 26 = 0.035435

Index for 27 = 0.0335711

Index for 28 = 0.0532554

Index for 29 = 0.0335503

Index for 30 = 0.0350808

KEY: поснбковандрей

**Код програми**

#include <iostream>

#include <fstream>

#include <math.h>

#include <string>

#include <map>

using namespace std;

string alphabet = "абвгдежзийклмнопрстуфхцчшщъыьэюя"; // без пробела

string keygood;

int count\_letter = 0;

int count\_vzlom = 0;

int letter\_max = 0;

double index\_sum;

double index(int size, string txt);

int ciphers(char\* txtq, string k2, int size);

int vzlomm(string text, int a);

int key(int size, string txt);

int main()

{

setlocale(LC\_ALL, "Russian");

string k2 = "до";

string k3 = "дух";

string k5 = "каска";

string k9 = "аллегория";

string k12 = "проспектлень";

string k15 = "скуратовильяста";

string k19 = "ялюблюсвоюмамуочень";

char file[] = "C:\\2.txt"; // текст

char vzlom[] = "C:\\1.txt"; // текст

ifstream go(vzlom);

while (!go.eof())

{

go.get();

count\_vzlom++;

}

go.close();

char\* vzl = new char[count\_vzlom];

for (int i = 0; i < count\_vzlom; i++)

{

vzl[i] = NULL; // заполняем нулями

}

ifstream goo(vzlom);

int q = 0;

while (!goo.eof())

{

goo.get(vzl[q]);

q++;

}

goo.close();

string u(vzl);

ifstream in(file);

while (!in.eof())

{

in.get();

count\_letter++;

}

in.close();

ifstream inn(file);

char\* mass = new char[count\_letter];

for (int i = 0; i < count\_letter; i++)

{

mass[i] = NULL; // заполняем нулями

}

ifstream in2(file);

int i = 0;

while (!in2.eof())

{

char letter;

in2.get(mass[i]);

mass[i] = tolower(mass[i]); // в строчные

if (alphabet.find(mass[i]) != -1)

{

i++;

}

else {

mass[i] = NULL;

count\_letter--;

}

}

in2.close();

char\* mass2 = new char[count\_letter];

for (int i = 0; i < count\_letter; i++)

{

mass2[i] = NULL;

}

for (int i = 0; i < count\_letter; i++)

{

mass2[i] = mass[i];

}

string aa(mass2);

cout << "Index for open text" << " = " << index(count\_letter, aa) << endl<<endl;

cout << "index for k2 : ";

ciphers(mass2, k2, count\_letter);

cout << "index for k3 : ";

ciphers(mass2, k3, count\_letter);

cout << "index for k5 : ";

ciphers(mass2, k5, count\_letter);

cout << "index for k9 : ";

ciphers(mass2, k9, count\_letter);

cout << "index for k12 : ";

ciphers(mass2, k12, count\_letter);

cout << "index for k15 : ";

ciphers(mass2, k15, count\_letter);

cout << "index for k19 : ";

ciphers(mass2, k19, count\_letter);

cout << endl << endl;

int a = 2;

while (a < 31)

{

vzlomm(u, a);

a++;

}

cout << endl << endl << "KEY: " << keygood<<endl;

return 0;

}

double index(int size, string txt) {

setlocale(LC\_ALL, "Russian");

double sum = 0;

int p = 0;

int mo = -1;

char\* monogr = new char[size];

double index;

for (int i = 0; i < size; i++) {

mo++;

monogr[mo] = NULL;

int letter = 0;

monogr[mo] = txt[i];

if (monogr[mo] != NULL) {

for (int j = i; j < size; j++) {

if (txt[j] == monogr[mo]) {

letter++;

}

}

index = (double(letter) \* (double(letter) - 1)) / (double(size) \* (double(size) - 1));

sum += index; //суммируем

for (int z = i; z < size; z++) {

if (txt[z] == monogr[mo]) txt[z] = NULL;

}

}

}

return sum;

}

int ciphers (char\* txtq, string shifr, int size)

{

int l = std::size(shifr);

int\* mass = new int[size];

int\* mass2 = new int[l];

int\* mass3 = new int[size];

string shifro;

char\* k\_ciphers = new char[size];

for (int i = 0; i < size; i++)

{

k\_ciphers[i] = '\0'; // заполняем нулями

}

setlocale(LC\_ALL, "Russian");

// массив с нашим текстом нумерация

for (int i = 0; i < size; i++)

{

int j = 0;

while (txtq[i] != alphabet[j])

{

j++;

}

mass[i] = j;

// cout << mass[i]<<" ";

}

//массив с ключем нумерация

for (int i = 0; i < l; i++)

{

int j = 0;

while ( shifr[i] != alphabet[j])

{

j++;

}

mass2[i] = j;

//cout << mass2[i] << " ";

}

// массив шифрованных букв

for (int i = 0; i < size; i++)

{

mass3[i] = (mass[i] + mass2[i%l]) % 32;

//cout << mass3[i]<<" ";

}

// массив с нашим текстом нумерация

for (int i = 0; i < size; i++)

{

int k = 0;

k = mass3[i];

k\_ciphers[i] = alphabet[k];

//cout << k\_ciphers[i] << " ";

}

k\_ciphers[size] = '\0';

string gg(k\_ciphers);

//cout << k\_ciphers;

cout<<index(size, gg)<<endl;

return 0;

}

int vzlomm(string text, int a)

{

setlocale(LC\_ALL, "Russian");

map<int, string>z;

for (int i = 0; i < a; i++)

{

z.emplace(i, "0");

}

int b = 0;

int j = 0;

int v;

int size = text.length();

string stud;

//cout << size<<endl;

for (int q = 0; q < a; q++)

{

stud.erase();

for (int i = q; i < size; i+=a)

{

if (text[i] != '\0')

{

stud.push\_back(text[i]);

}

}

z.at(q) = stud;

}

int r = 0;

//////////////////

double sum = 0 ;

for (int i = 0; i < a; i++)

{

r = z.at(i).length();

sum+=index(r, z.at(i));

//cout << "Index for " << a << " = " << index(r, z.at(i)) << endl;

}

cout<<"Index for "<<a<<" = "<< sum / a<<endl;

if (a == 14)

{

for (int i = 0; i < a; i++)

{

int n = z.at(i).length();

key(n, z.at(i));

}

}

return 0;

}

int key(int size, string txt)

{

int count\_max = 0;

int count;

int mesto;

for (int j = 0; j < 32; j++)

{

count = 0;

for (int i = 0; i < size; i++)

{

if (txt[i] == alphabet[j])

{

count++;

}

}

if (count > count\_max)

{

count\_max = count;

mesto = j;

}

}

//cout << mesto << endl << endl;

int o\_mesto = 14;

int otvet\_o = 0;

otvet\_o = abs(o\_mesto - mesto);

keygood.push\_back(alphabet[otvet\_o]);

return 0;

}

***Висновки:***

Засвоєння методів частотного криптоаналізу. Здобуття навичок роботи та аналізу потокових шифрів гамування адитивного типу на прикладі шифру Віженера.