Akademia Nauk Stosowanych w Nowym Sączu Teoretyczne i technologiczne podstawy multimediów - laboratorium			
Temat: Algorytm szyfrujący			**Symbol: TiTPM_L2**
Nazwisko i imię:		**Ocena sprawozdania**	**Zaliczenie:**
1. Szczepanek Piotr			
Data wykonania ćwiczenia: 04.10.2022	**Grupa: L2**		

RSA

Algorytm Rivesta-Shamira-Adlemana – jeden z pierwszych i obecnie najpopularniejszych asymetrycznych algorytmów kryptograficznych z kluczem publicznym, zaprojektowany w 1977 przez Rona Rivesta, Adiego Shamira oraz Leonarda Adlemana. Pierwszy algorytm, który może być stosowany zarówno do szyfrowania, jak i do podpisów cyfrowych. Bezpieczeństwo szyfrowania opiera się na trudności faktoryzacji dużych liczb złożonych. Jego nazwa pochodzi od pierwszych liter nazwisk jego twórców.

```cpp
#include<iostream>
#include<stdlib.h>
#include<math.h>
#include<string.h>

using namespace std;

int a, b, c, t, i, flag;
long int e[50], d[50], temp[50], j;
char en[50], m[50];
char wiadomosc[100];
int prime(long int);
void klucz_szyfrujacy();
long int cd(long int);
void szyfrowanie();
void deszyfrowanie();

int main()
{
    cout << endl << "Podaj liczbe pierwsza: ";
    cin >> a;

    //sprawdzamy czy ta liczba jest pierwsza czy tez nie
    flag = prime(a);
    if (flag == 0)
    {
        cout << endl << "Podana liczba nie jest liczba pierwsza!!!";
        exit(0);
    }

    cout << endl << "Podaj kolejna liczbe pierwsza: ";
    cin >> b;

    flag = prime(b);
    if (flag == 0 || a == b)
    {
        cout << endl << "Podana liczba nie jest liczba pierwsza!!!";
        exit(0);
    }

    cout << endl << "Podaj wiadomosc do zaszyfrowania: ";
    cin >> wiadomosc;

    for (i = 0; wiadomosc[i] != NULL; i++)
        m[i] = wiadomosc[i];
    c = a * b;
    t = (a - 1) * (b - 1);

    klucz_szyfrujacy();
    cout << endl << "Mozliwe wartosci wykladnika prywatnego i publicznego to: ";

    for (i = 0; i < j - 1; i++)
        cout << "\c" << e[i] << "\t" << d[i];

    szyfrowanie();
    deszyfrowanie();
    return 0;
}

int prime(long int pr)
{
    int i;
    j = sqrt(pr);
    for (i = 2; i <= j; i++)
    {
        if (pr % i == 0)
            return 0;
    }
    return 1;
}

//deszyfrowanie
void klucz_szyfrujacy()
{
    int k;
    k = 0;
    for (i = 2; i < t; i++)
    {
        if (t % i == 0)
            continue;
        flag = prime(i);
```

```
        if (flag == 1 && i != a && i != b)
        {
            e[k] = i;
            flag = cd(e[k]);
            if (flag > 0)
            {
                d[k] = flag;
                k++;
            }
            if (k == 99)
                break;
        }
    }
}

long int cd(long int a)
{
    long int k = 1;
    while (1)
    {
        k = k + t;
        if (k % a == 0)
            return(k / a);
    }
}

//zaszyfrowanie
void szyfrowanie()
{
    long int pt, ct, key = e[0], k, len;
    i = 0;
    len = strlen(wiadomosc);

    while (i != len)
    {
        pt = m[i];
        pt = pt - 96;
        k = 1;
        for (j = 0; j < key; j++)
        {
            k = k * pt;
            k = k % c;
        }
        temp[i] = k;
        ct = k + 96;
        en[i] = ct;
        i++;
    }
    en[i] = -1;
    cout << endl << "Zaszyfrowana wiadomosc ma postac: ";
    for (i = 0; en[i] != -1; i++)
        cout << en[i];
}

//odszyfrowywaniea
void deszyfrowanie()
{
    long int pt, ct, key = d[0], k;
    i = 0;
    while (en[i] != -1)
    {
        ct = temp[i];
        k = 1;
        for (j = 0; j < key; j++)
        {
            k = k * ct;
            k = k % c;
        }
        pt = k + 96;
        m[i] = pt;
        i++;
    }
    m[i] = -1;
    cout << endl << "Rozszyfrowana wiadomosc ma postac: ";
    for (i = 0; m[i] != -1; i++)
        cout << m[i];

    cout << endl;
}
```

Rysunek 1 Kod programu

```
Podaj liczbe pierwsza: 13

Podaj kolejna liczbe pierwsza: 7

Podaj wiadomosc do zaszyfrowania: programik

Mozliwe wartosci wykladnika prywatnego i publicznego to:
5       29
11      59
17      17
19      19
23      47
29      5
31      7
Zaszyfrowana wiadomosc ma postac: ¬îžčîam▒ξ
Rozszyfrowana wiadomosc ma postac: programik
```

Rysunek 2 Działanie programu

Jak działa powyższy program?

1. Tworzenie kluczy

Na samym początku Użytkownik musi podać dwie liczby pierwsze. Oznaczyliśmy je jako a i b. Program sprawdza, czy liczby, które podał Użytkownik są pierwsze, czy też nie.

Następnie oblicza c = a * b, gdzie c jest modułem klucza prywatnego i publicznego.

Następnie obliczamy funkcję Eulera ø (c) = (a − 1)(b – 1.
Wybieramy taką liczbę całkowitą e, że e jest względnie pierwsza do
ø(c) i 1 < e < ø(c).
Liczba e jest tutaj wykładnikiem klucza publicznego, używanym do szyfrowania.

Wyznaczamy następnie wykładnik prywatny, który ma być odwrotnością modulo Ø liczby e, czyli
d · e mod ø (c) = 1.

2. Szyfrowanie wiadomości

Wiadomości są szyfrowane za pomocą wygenerowanego klucza publicznego i są znane wszystkim.

Klucz publiczny jest funkcją zarówno e, jak i c.

Jeśli przyjmiemy, że M jest wiadomością w postaci zwykłego tekstu, to jej zaszyfrowana postać wygląda następująco:

C = M^ c (mod c)

3. Odszyfrowywanie wiadomości

Klucz prywatny jest funkcją zarówno d, jak i c.

Jeśli C jesttekstem zaszyfrowanym, to zwykły odszyfrowany tekst M ma postać:

M = C^ d (mod c).