# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ТАРАСА ШЕВЧЕНКА



## ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №13**

# з курсу

**«Безпека мереж і комп’ютерних систем»**

*Студента 2 курсу*

*групи ПП-21 спеціальності 122 «Комп'ютерні науки» ОП «Прикладне програмування»*

%username%

*Викладач:*

проф. Сайко В.Г.

## Київ – 202

**1.Назва роботи**

Криптографічні алгоритми, які використовуються для формування електронно-цифроового підпису (ЕЦП) підпису. Формування ЕЦП за допомогою алгоритму DSA.

1. **Тема роботи**

Криптографічні алгоритми, які використовуються для формування електронно-цифроового підпису (ЕЦП) підпису. Формування ЕЦП за допомогою алгоритму DSA.

1. **Мета роботи**

Ознайомитися з основними вимогами до ЕЦП. Ознайомитися з принципом функціонування алгоритму DSA.

1. **Умова завдання**



1. **Рішення**

Код на С#:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab13\_DSA\_

{

class Program

{

public static int NOD(int a, int b)

{

if (a == b)

return a;

else

if (a > b)

return NOD(a - b, b);

else

return NOD(b - a, a);

}

public static int fast(int a, int r, int n)

{

int a1 = a;

int z1 = r;

int x = 1;

while (z1 != 0)

{

while (z1 % 2 == 0)

{

z1 /= 2;

a1 = (a1 \* a1) % n;

}

z1 -= 1;

x = (x \* a1) % n;

}

return x;

}

public static int hash(string s, int n)

{

char[] b = new char[26];

b[0] = 'a'; b[1] = 'b'; b[2] = 'c'; b[3] = 'd'; b[4] = 'e'; b[5] = 'f'; b[6] = 'g'; b[7] = 'h'; b[8] = 'i'; b[9] = 'j'; b[10] = 'k';

b[11] = 'l'; b[12] = 'm'; b[13] = 'n'; b[14] = 'o'; b[15] = 'p'; b[16] = 'q'; b[17] = 'r'; b[18] = 's'; b[19] = 't'; b[20] = 'u'; b[21] = 'v';

b[22] = 'w'; b[23] = 'x'; b[24] = 'y'; b[25] = 'z';

int k = 150;

int f = 0;

s = s.ToLower();

for (int i = 0; i < s.Length; i++)

{

for (int j = 0; j < 26; j++)

{

if (b[j] == s[i])

{

f = j + 1;

}

}

k = Convert.ToInt32(Math.Pow(k + f, 2) % n);

}

return k;

}

static void Main(string[] args)

{

Console.WriteLine("Enter keygen method:\n1 - automatic, 2 - manual");

string ind = Console.ReadLine();

Console.WriteLine("Enter the plaintext:");

string M = Console.ReadLine();

int q;

if (ind == "2") { Console.WriteLine("Enter q"); q = int.Parse(Console.ReadLine()); }

else { Random rdm = new Random(); q = rdm.Next(10, 100); }

int p = q + 1;

while (true)

{

bool f = false;

for (int i = 2; i < p - 1; i++)

{

if (p % i == 0)

{

f = true;

break;

}

}

if (!f && (p - 1) % q == 0)

{

break;

}

else

{

p++;

}

}

double g = 0;

Random rand = new Random();

while (g < 1)

{

g = Math.Pow(rand.Next(1, p - 1), (p - 1) / q);

}

int x = rand.Next(0, q);

int y = fast(Convert.ToInt32(g), x, p);

int s = 0;

int r = 0;

while (true)

{

int k = rand.Next(0, q);

r = fast(Convert.ToInt32(g), k, p) % q;

int k1 = 0;

while ((k1 \* k) % q != 1)

{

k1++;

}

s = Convert.ToInt32(k1 \* (hash(M, q) + x \* r)) % q;

if (r != 0 || s != 0)

{

Console.WriteLine("Message with signature: [{0}, {1}, {2}]", M, r, s);

break;

}

}

Console.WriteLine("Signature test:\nEnter a message to test the signature:");

M = Console.ReadLine();

int s1 = 0;

while ((s1 \* s) % q != 1)

{

s1++;

}

int w = s1 % q;

int u1 = (hash(M, q) \* w) % q;

int u2 = (r \* w) % q;

double mp1 = fast(Convert.ToInt32(g), u1, p);

double mp2 = fast(y, u2, p);

double res = mp1 \* mp2;

res %= p;

res %= q;

int v = Convert.ToInt32(res);

Console.WriteLine("v = {0}", v);

if (v == r)

{

Console.WriteLine("Values are identical, DSA is correct: {0} = {1}", v, r);

}

else

{

Console.WriteLine("Values are not identical, DSA is not correct: {0} != {1}", v, r);

}

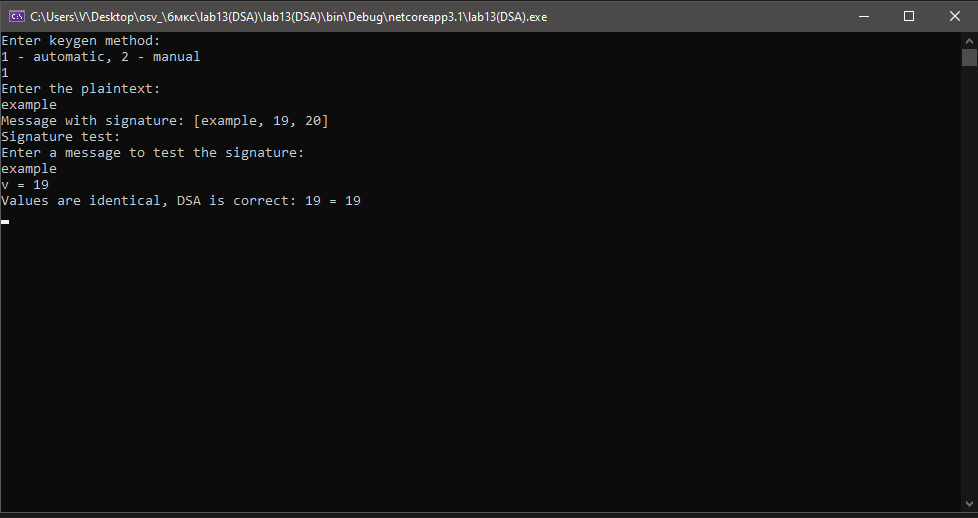
Console.ReadLine();

}

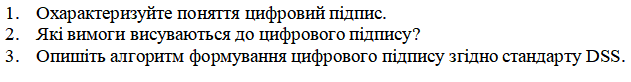
}

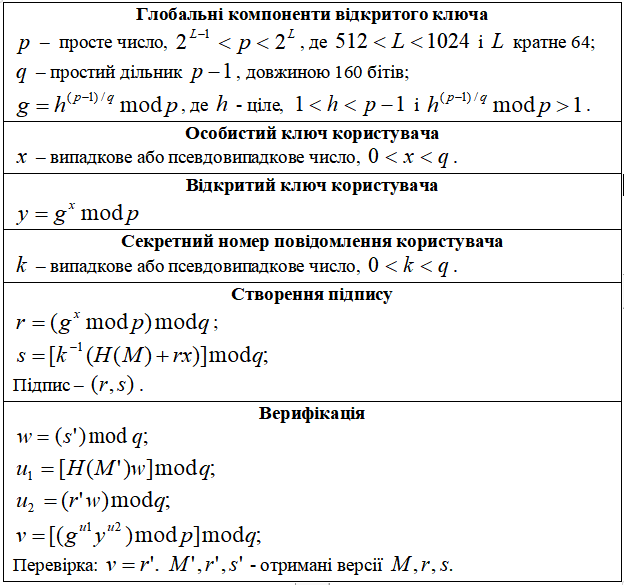
}

Результат роботи:



**Контрольні запитання:**



1. Це стандарт обробки інформації, прийнятий Національним інститутом стандартів і технологій, що використовує алгоритм DSА
2. Щоб створити підпис, користувач обчислює дві величини і , які являються функціями компонентів відкритого ключа , особистого ключа користувача , хеш-коду повідомлення , обчисленого згідно алгоритму SHA-1, і деякого цілого числа , яке повинно вибиратися випадковим або псевдовипадковим чином і бути унікальним для кожного виконання підпису.
3. 

**Висновки**

В результаті виконання даної лабораторної роботи я ознайомився з основними вимогами до ЕЦП, принципом функціонування алгоритму DSA та навчився проводити верифікацію. Ознайомився з математичними принципами функціонування алгоритму DSA, з формуванням електронного підпису. Створив програмний засіб, що реалізує верифікацію DSA, використовуючи мову програмування С#. Створив методи у функції main, викликав їх, отримав та оцінив результат.

Вважаю дану лабораторну роботу виконаною в повному обсязі.