|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** ***ИУК «Информатика и управление»***

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ЛАБОРАТОРНАЯ РАБОТА №4**

**«Однонаправленные хэш-функции. Электронная цифровая подпись»**

**ДИСЦИПЛИНА: «Защита информации»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-72Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Карельский М.К. )  (Подпись) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Ерохин И.И. )  (Подпись) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2023

**Цель:** изучить различные алгоритмы однонаправленного хэширования данных, основанные на симметричных блочных алгоритмах шифрования. Ознакомиться со схемами цифровой подписи и получить навыки создания и проверки подлинности электронной цифровой подписи.

**Задачи:** изучить предложенный теоретический материал для получения информации о понятии, параметрах, схемах однонаправленных хэш-функций и ЭЦП. Ознакомиться с принципом действия алгоритма Эль-Гамаля. Реализовать приложение, позволяющее вычислять и проверять ЭЦП, сформированную по алгоритмам RSA и Эль-Гамаля. Протестировать правильность работы разработанного приложения. Для заданных в варианте открытых ключей пользователя проверить подлинность подписанных по алгоритму RSA хэш-значений, для алгоритма Эль-Гамаля найти открытый ключ и построить подпись для хэш-значения. Произвести проверку подписи.

**Задание:**

1. Реализовать приложение, позволяющее вычислять и проверять ЭЦП, сформированную по алгоритмам RSA и Эль-Гамаля.
2. С помощью реализованного приложения выполнить следующие задания:
   1. Протестировать правильность работы разработанного приложения.
   2. Для заданных в варианте открытых ключей пользователя проверить подлинность подписанных по алгоритму RSA хэш-значений m некоторых сообщений M.
   3. Абоненты некоторой сети применяют подпись Эль-Гамаля с известными общими параметрами p и g. Для указанных в варианте секретных параметров абонентов найти открытый ключ и построить подпись для хэш-значения m некоторого сообщения M. Проверить правильность подписи

Для построения подписи Эль-Гамаля следует использовать открытые параметры p = 23, g = 5.

**Вариант 7**

* ЭЦП по алгоритму RSA:
  + Открытые ключи:
    - n = 221
    - e = 43
  + Проверяемые сообщения (m, s):
    - (59, 19)
    - (79, 164)
    - (58, 20)
* ЭЦП по алгоритму Эль-Гамаля:
  + Секретные параметры:
    - x = 14
    - k = 17
  + m = 14

**Листинг:**

***LW4\_1.py***

def fast\_pow(x, y):

    if y == 0:

        return 1

    if y == -1:

        return 1. / x

    p = fast\_pow(x, y // 2)

    p \*= p

    if y % 2:

        p \*= x

    return p

def encode(message, e, n):

    return fast\_pow(message, e) % n

def decode(message, d, n):

    return fast\_pow(message, d) % n

n = 221

e = 43

check = [[59, 19], [79, 164], [58, 20]]

for i in range(0, 3):

    if encode(check[i][0], e, n) == check[i][1]:

        print(f'({check[i][0]}, {check[i][1]}): success')

    else:

        print(f'({check[i][0]}, {check[i][1]}): failed')

***LW4\_2.py***

import math

def fast\_pow(x, y):

    if y == 0:

        return 1

    if y == -1:

        return 1. / x

    p = fast\_pow(x, y // 2)

    p \*= p

    if y % 2:

        p \*= x

    return p

def reverse\_element(f, d):

    X = [1, 0, f]

    Y = [0, 1, d]

    while True:

        if Y[2] == 0:

            print("Error")

            return

        elif Y[2] == 1:

            return Y[1]

        else:

            q = X[2]//Y[2]

            t = [0, 0, 0]

            for i in range(0, 3):

                t[i] = X[i] - q\*Y[i]

                X[i] = Y[i]

                Y[i] = t[i]

p = 23

g = 5

x = 14

k = 17

m = 14

y = math.pow(g, x) % p

a = math.pow(g, k) % p

f = p - 1

kr = reverse\_element(f, k)

b = (kr \* (m - x \* a)) % f

if ((fast\_pow(y, a)\*fast\_pow(a, b)) % p) == (fast\_pow(g, m) % p):

    print("Success")

else:

    print("Failed")

**Результат:**



**Рис. 1.** RSA



**Рис. 2.** Эль-Гамаль

**Вывод:** в ходе выполнения лабораторной работы были изучены различные алгоритмы однонаправленного хэширования данных, основанные на симметричных блочных алгоритмах шифрования, схемы цифровой подписи, получены навыки создания и проверки подлинности электронной цифровой подписи.