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 _____ (_____Калашников А.С.__)
(Подпись) (Ф.И.О.)

Проверил: _____ (____Ерохин И.И.___)
(Подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Калуга, 2023

Целью выполнения лабораторной работы изучение различных алгоритмов однонаправленного хэширования данных, основанные на симметричных блочных алгоритмах шифрования. Ознакомление со схемами цифровой подписи и получение навыков создания и проверки подлинности электронной цифровой подписи.

Основными задачами выполнения лабораторной работы являются:

1. Изучить предложенный теоретический материал для получения информации о понятии, параметрах, схемах однонаправленных хэш-функций и ЭЦП.
2. Ознакомиться с принципом действия алгоритма Эль-Гамаля Реализовать приложение, позволяющее вычислять и проверять ЭЦП, сформированную по алгоритмам RSA и Эль-Гамаля.
3. Протестировать правильность работы разработанного приложения.
4. 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.
5. Произвести проверку подписи.

Результатами работы являются:

Разработанная программа согласно варианту задания

Подготовленный отчет

Вариант № 6

Задание:

1. Реализовать приложение, позволяющее вычислять и проверять ЭЦП, сформированную по алгоритмам RSA и Эль-Гамаля.

2. С помощью реализованного приложения выполнить следующие задания:

2.1. Протестировать правильность работы разработанного приложения.

2.2. Для заданных в варианте открытых ключей пользователя проверить подлинность подписанных по алгоритму RSA хэш-значений m некоторых сообщений M.

2.3. 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 $p = 23$, $g = 5$.

6	$n = 143,\ \ e = 37$	$\langle 46, 85\rangle,\quad \langle 16, 74\rangle,\quad \langle 129, 116\rangle$	$x = 19,\quad k = 5$	$m = 11$

Листинг программы:

1)rsa

```
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 = 143
e = 37
check = [[46, 85], [16, 74], [129, 116]]
for i in range(0, 3):
    if encode(check[i][0], e, n) == check[i][1]:
        print(" Проверка набора номер ", i + 1, " пройдена успешно")
    else:
        print(" Проверка набора номер ", i + 1, " не пройдена")
```

2) Эль-Гамаль

```
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("Нет обратного элемента")
            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 = 19
k = 5
m = 11
```

```
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("Успешно")
else:
    print("Не успешно")
```

Результат выполнения программы:

```
Проверка набора номер  1  не пройдена
Проверка набора номер  2  не пройдена
Проверка набора номер  3  не пройдена
```

Рис.1. Результат работы rsa

```
Не успешно
```

Рис.2. Результат работы Эль-Гамаль

Вывод: в результате выполнения данной лабораторной работы были изучены различные алгоритмы однонаправленного хэширования данных, основанные на симметричных блочных алгоритмах шифрования, схемах цифровой подписи. Получены навыки создания и проверки подлинности электронной цифровой подписи.