|  |  |
| --- | --- |
| **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 №3**

**«Алгоритм RSA. Обмен ключами симметричных алгоритмов с использованием ассиметричных криптосистем»**

**ДИСЦИПЛИНА: «Защита информации»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил: студент гр. ИУК4-72Б | |  |  | ( | Сафронов Н.С. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |
| Проверил: | |  |  | ( | Ерохин И.И. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |

|  |  |
| --- | --- |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | |
|  | - Балльная оценка:  - Оценка: |

Калуга, 2023

**Цель работы:** ознакомиться с математическими принципами функционирования алгоритма RSA, научиться шифрование/дешифрование с помощью данного алгоритма, ознакомиться с принципом реализации обмена ключами с использованием схемы Диффи-Хеллмана, освоить данный алгоритм обмена ключами.

**Постановка задачи**

1. Рассмотреть общие математические принципы организации процедуры шифрования/дешифрования при использовании метода RSA.
2. Рассмотреть схему обмена ключами по алгоритму Диффи-Хеллмана.
3. Реализовать программно алгоритм шифрования и дешифрования методом RSA.
4. Провести шифрование открытого текста, выбранного согласно варианту, указанному преподавателем, и его последующее восстановление.
5. Рассмотреть схему Диффи-Хеллмана с общим простым числом q и первообразным корнем a. Вами выбран секретный ключ . При обмене ключами с вашим респондентом, имеющим открытый ключ , вы получили от него общий секретный ключ К. Состоялся ли обмен ключами? Обоснуйте ответ. Вычислите значение открытого ключа .

Значения параметров q, a, , , К выбрать согласно варианту.

**Вариант 14**

Слово – самозагрузка.

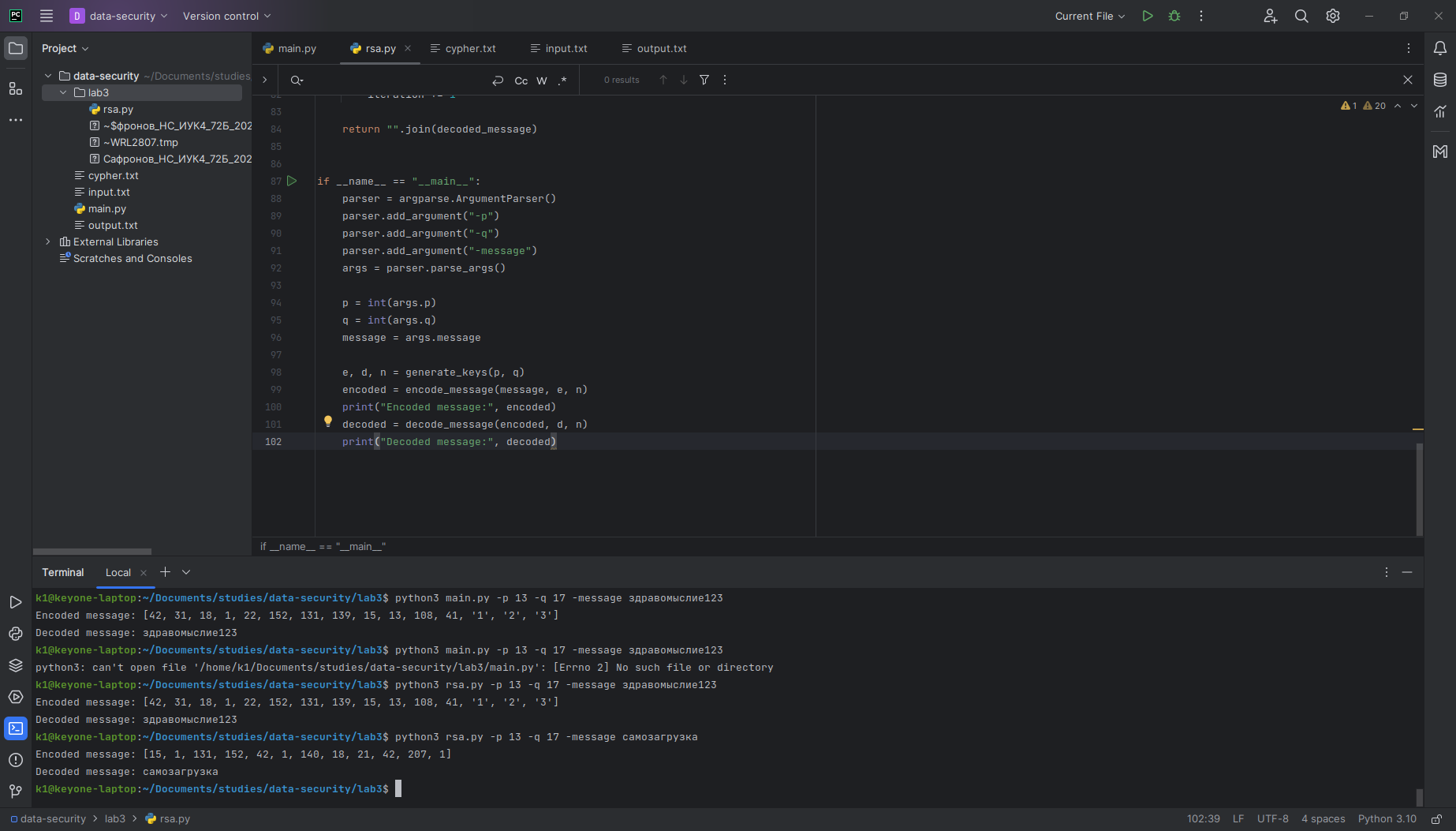
**Ход выполнения работы**

**Алгоритм шифрования и дешифрования методом RSA**

**Листинг программы**

# RSA  
  
import argparse  
import math  
import typing as \_t  
  
RUSSIAN\_ALPHABET = [  
 'а', 'б', 'в', 'г', 'д', 'е', 'ё', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н',  
 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь',  
 'э', 'ю', 'я'  
]  
  
  
def fast\_pow(x: int, y: int) -> float:  
 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 generate\_keys(p: int, q: int) -> \_t.Tuple[int, int, int]:  
 n = p \* q  
 euler = (p - 1) \* (q - 1)  
  
 e = 0  
 i = 2  
 while i < euler:  
 e = math.gcd(euler, i)  
 if e == 1:  
 e = i  
 break  
 i += 1  
  
 d = 0  
 i = 2  
 while i < n:  
 if (i \* e) % euler == 1:  
 d = i  
 break  
 i += 1  
  
 return e, d, n  
  
  
def encode\_number(number: int, e: int, n: int) -> float:  
 return fast\_pow(number, e) % n  
  
  
def decode\_number(number: int, d: int, n: int) -> float:  
 return fast\_pow(number, d) % n  
  
  
def encode\_message(message: str, e: int, n: int) -> list:  
 iteration = 0  
 encoded\_message: list = [None] \* len(message)  
  
 for letter in message:  
 try:  
 index = RUSSIAN\_ALPHABET.index(letter) + 1  
 encoded\_message[iteration] = encode\_number(index, e, n)  
 except ValueError:  
 encoded\_message[iteration] = letter  
 iteration += 1  
  
 return encoded\_message  
  
  
def decode\_message(message: list, d: int, n: int) -> str:  
 iteration = 0  
 decoded\_message: list = [""] \* len(message)  
  
 for letter in message:  
 try:  
 current = decode\_number(letter, d, n)  
 decoded\_message[iteration] = RUSSIAN\_ALPHABET[current - 1]  
 except TypeError:  
 decoded\_message[iteration] = letter  
 iteration += 1  
  
 return "".join(decoded\_message)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 parser = argparse.ArgumentParser()  
 parser.add\_argument("-p")  
 parser.add\_argument("-q")  
 parser.add\_argument("-message")  
 args = parser.parse\_args()  
  
 p = int(args.p)  
 q = int(args.q)  
 message = args.message  
  
 e, d, n = generate\_keys(p, q)  
 encoded = encode\_message(message, e, n)  
 print("Encoded message:", encoded)  
 decoded = decode\_message(encoded, d, n)  
 print("Decoded message:", decoded)

**Результат выполнения программы**



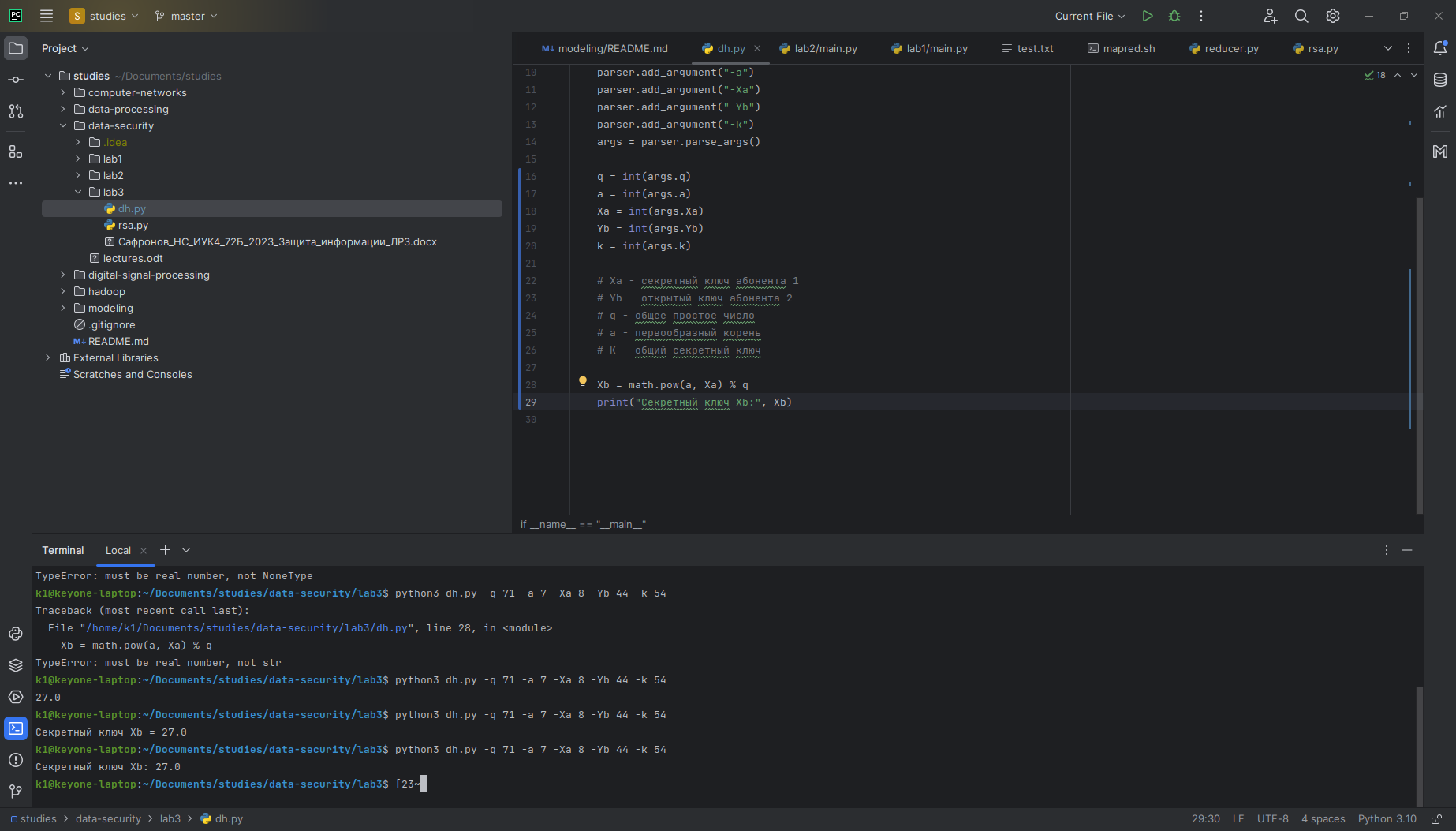
**Рисунок 1** – Результат шифрования и дешифрования алгоритмом RSA

**Схема Диффи-Хеллмана**

**Листинг программы**

# Диффи-Хеллман  
  
import argparse  
import math  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 parser = argparse.ArgumentParser()  
 parser.add\_argument("-q")  
 parser.add\_argument("-a")  
 parser.add\_argument("-Xa")  
 parser.add\_argument("-Yb")  
 parser.add\_argument("-k")  
 args = parser.parse\_args()  
  
 q = int(args.q)  
 a = int(args.a)  
 Xa = int(args.Xa)  
 Yb = int(args.Yb)  
 k = int(args.k)  
  
 # Xa - секретный ключ абонента 1  
 # Yb - открытый ключ абонента 2  
 # q - общее простое число  
 # a - первообразный корень  
 # K - общий секретный ключ  
  
 Xb = math.pow(a, Xa) % q  
 print("Секретный ключ Xb:", Xb)

**Результат выполнения программы**



**Рисунок 2** – Полученный секретный ключ по схеме Диффи-Хеллмана

**Вывод**: в ходе выполнения лабораторной работы были получены навыки шифрования и дешифрования с помощью RSA, был изучен принцип реализации обмена ключами с использованием схемы Диффи-Хеллмана.