



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  $X_A$ . При обмене ключами с вашим респондентом, имеющим открытый ключ  $Y_B$ , вы получили от него общий секретный ключ  $K$ . Состоялся ли обмен ключами? Обоснуйте ответ. Вычислите значение открытого ключа  $Y_A$ .

Значения параметров  $q$ ,  $a$ ,  $X_A$ ,  $Y_B$ ,  $K$  выбрать согласно варианту.

### **Вариант 14**

Слово – самозагрузка.

$$q = 71, a = 7, X_A = 8, Y_B = 44, K = 54$$

### **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

```

PS D:\Dev\bmstu-7th-term\data-security\lab3> python .\rsa.py -p 13 -q 17 -message самозагрузка
Encoded message: [15, 1, 131, 152, 42, 1, 140, 18, 21, 42, 207, 1]
Decoded message: самозагрузка

```

**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()

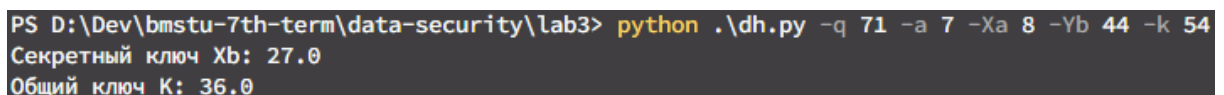
    q = int(args.q)
    a = int(args.a)
    Xa = int(args.Xa)
    Yb = int(args.Yb)
    k = int(args.k)

    # 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)

    Kb = math.pow(Yb, Xa) % q
    print("Общий ключ K:", Kb)
```

### Результат выполнения программы



```
PS D:\Dev\bmstu-7th-term\data-security\lab3> python .\dh.py -q 71 -a 7 -Xa 8 -Yb 44 -k 54
Секретный ключ Xb: 27.0
Общий ключ K: 36.0
```

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

Вычисленный ключ  $K = 36$  не равен данному в условии общему ключу  $K = 54$ , поэтому обмен ключами не состоялся.

**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.