УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №3

по предмету

Теория Информации

Вариант 1

Выполнил:

Горник А.

Проверил:

Болтак С.В.

Группа 351003

Минск 2025

1. **Пример работы алгоритма быстрого возведения в степень с использованием модульной арифметики.**

**315 mod 11 = ?  
a – основание(3)  
n – показатель степени(15)**

**e – четное n или нет(да/нет)**

**r – результат(вначале равен 1)**

**x – значение mod(11)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Шаг | a | n | e | Действие | r mod x |
| 1 | 3 | 15 | нет | r = 1 \* 3 mod 11 = 3 n = 15 - 1 = 14 | 3 |
| 2 | 3 | 14 | да | a = 3^2 mod 11 = 9 n = 14 / 2 = 7 | 3 |
| 3 | 9 | 7 | нет | r = 3 \* 9 mod 11 = 5 n = 7 - 1 = 6 | 5 |
| 4 | 9 | 6 | да | а = 9^2 mod 11 = 4 n =6 /2 = 3 | 5 |
| 5 | 4 | 3 | нет | r = 5 \* 4 mod 11 = 9 n = 3 - 1 = 2 | 9 |
| 6 | 4 | 2 | да | а = 4^2 mod 11 = 5 n =2 /2 = 1 | 9 |
| 7 | 5 | 1 | нет | r = 9 \* 5 mod 11 = 9 n = 1 - 1= 0 | 1 |

1. **Пример поиска всех первообразных корней для p и заданного модуля.**
2. Дано простое p = 37
3. Ищем простые делители p-1 = 36 = 2^2\* 3^2
4. Проверяем является ли число 2 первообразным корнем по модулю 37:

2^18 mod 37= 36; 2^12 mod 37 = 26. Число 2 является первообразным по модулю 37.

1. Проверяем является ли число 3 первообразным корнем по модулю 37:

3^18 mod 37= 1; 3^12 mod 37 = 10. Число 3 не является первообразным по модулю 37.

1. Если найден один первообразный корень g по модулю p, остальные корни имеют вид g^k, где НОД(k, p-1) = 1. Для p - 1 = 36, g = 2, допустимые k: 1,5,7,11,13,17,19,23,25,29,31,35:

2^1 mod 37 = 2

2^5 mod 37 = 32

2^7 mod 37 = 17

2^11 mod 37 = 13

2^13 mod 37 = 15

2^17 mod 37 = 18

2^19 mod 37 = 35

2^23 mod 37 = 5

2^25 mod 37 = 20

2^29 mod 37 = 24

2^31 mod 37 = 22

2^35 mod 37 = 19

Тогда все первообразные корни для модуля p = 37 это 2,5,13,15,17,18,19,20,22,24,32,35,

1. **Пример работы расширенного алгоритма Евклида** (числа не взаимно простые, студент должен привести пример с взаимно простыми числами)

**x1\*a + y1\*b = нод(a,b), a = 23, a1 = 17, (a,b) = 1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Итерация** | **q** | **a₀** | **a₁** | **x₀** | **x₁** | **y₀** | **y₁** |
| 0 | – | 23 | 17 | 1 | 0 | 0 | 1 |
| 1 | 1 | 17 | 6 | 0 | 1 | 1 | -1 |
| 2 | 2 | 6 | 5 | 1 | -2 | -1 | 3 |
| 3 | 1 | 5 | 1 | -2 | 3 | 3 | -4 |
| 4 | 5 | 1 | 0 | 3 | -17 | -4 | 23 |

x1 = 3, y1 = -4

3 \* 23 + (-4) \* 17 = 69 - 68 = 1

Начинаем обратный ход с предпоследней операции:  
1 = 5 – 2\*2;  
1 = 5 – 2 \* (127 – 5 \* 25); 1 = 5 \* 51 – 2 \* 127 (подставляем выражение для 2)

1 = (132 – 127) \* 51 – 2 \* 127; 1 = 132 \* 51 − 127 \* 53 (подставляем выражение для 5)

1 = 132 \* 51 – (391 – 132 \* 2) \* 53; 1 = 132 \* 157−391 \* 53 (подставляем выражение для 127)

1 = (523 – 391) \* 157−391 \* 53; 1 = 523 \* 157 – 391 \* 210 (подставляем выражение для 132)  
Получаем ответ: x = 157,y = −210

**Задание:**

Разработать программное средство, выполняющее вычисление открытого ключа **(*KO*)** алгоритма ***RSA*** и побайтовое шифрование данным ключом по алгоритму ***RSA*** произвольного файла. Значения параметров ***p,*** ***q*** и ***KС*,** а также имя входного файла задаются пользователем. Программа должна осуществлять проверку ограничений на вводимые пользователем значения параметров алгоритма. Организовать вывод содержимого зашифрованного файла на экран в виде чисел в 10 системе счисления.

Разработать программное средство, выполняющее расшифрование файла, каждый 16-битный блок которого представляет собой зашифрованное по алгоритму ***RSA*** 8-битное значение. Значения модуля ***r*** и ***закрытого*** ключа ***KС***задаются пользователем.

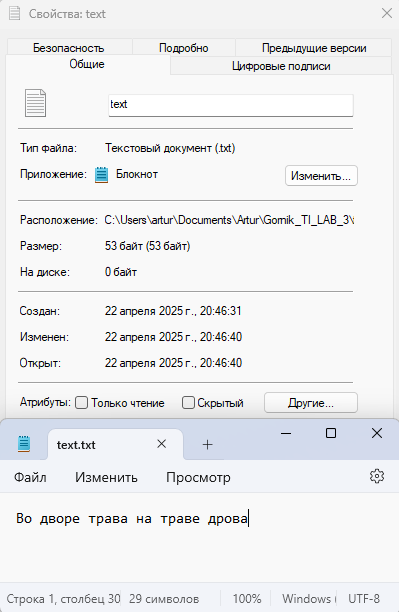
Использовать алгоритм быстрого возведения в степень и расширенный алгоритм Евклида.

При использовании длинной арифметики для определения простоты числа использовать один из вероятностных тестов: тест Ферма или тест Миллера-Рабина.

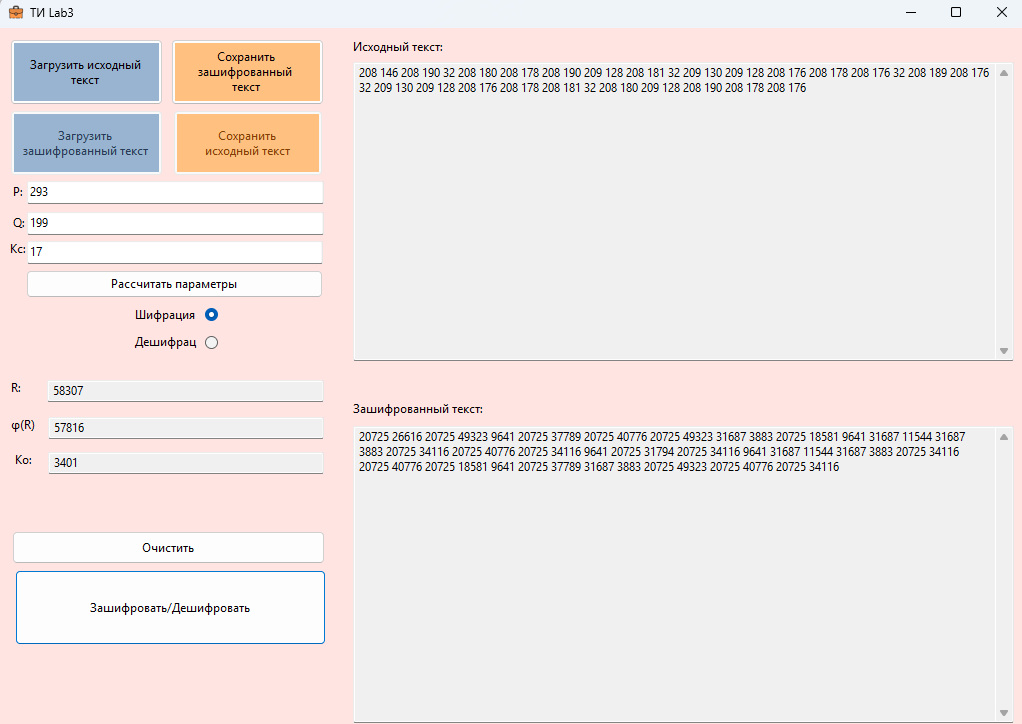
Результат работы программы – зашифрованный/расшифрованный файл/ы.

Тесты:

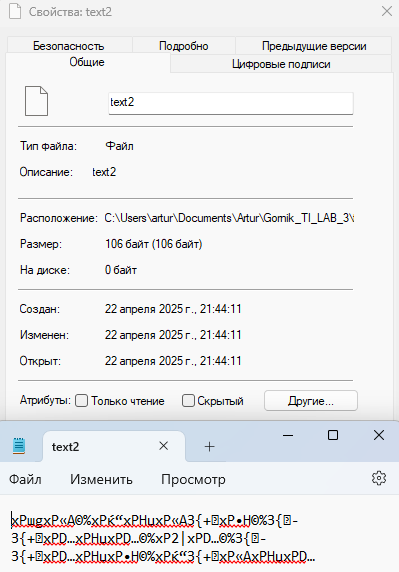
1. Шифрование текста:



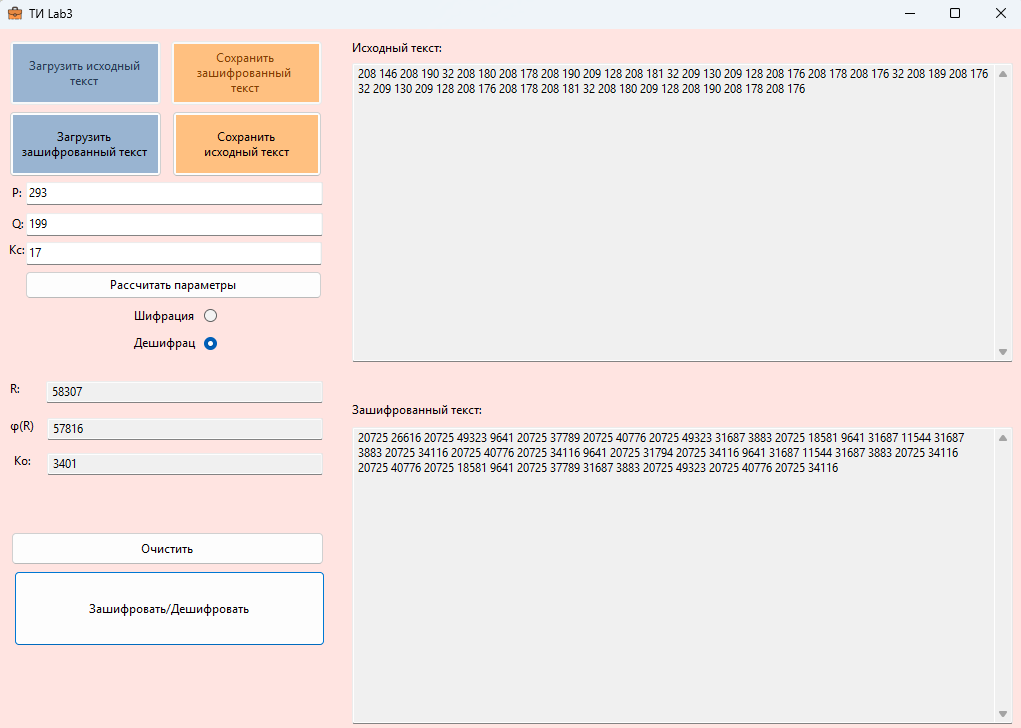
Результат работы шифратора:



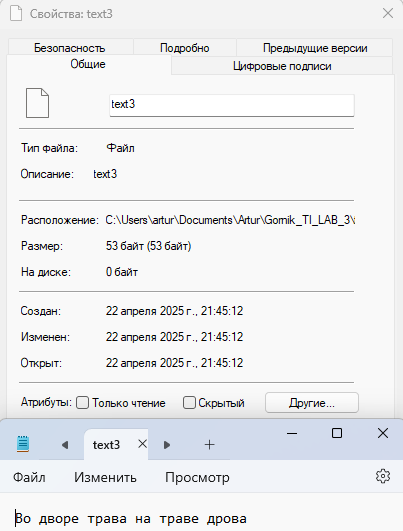
Зашифрованный текст:



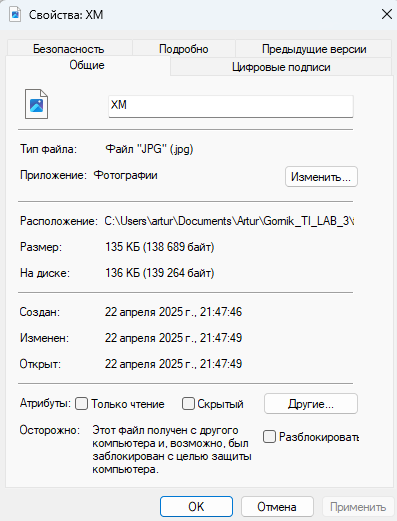
Результат работы дешифратора:

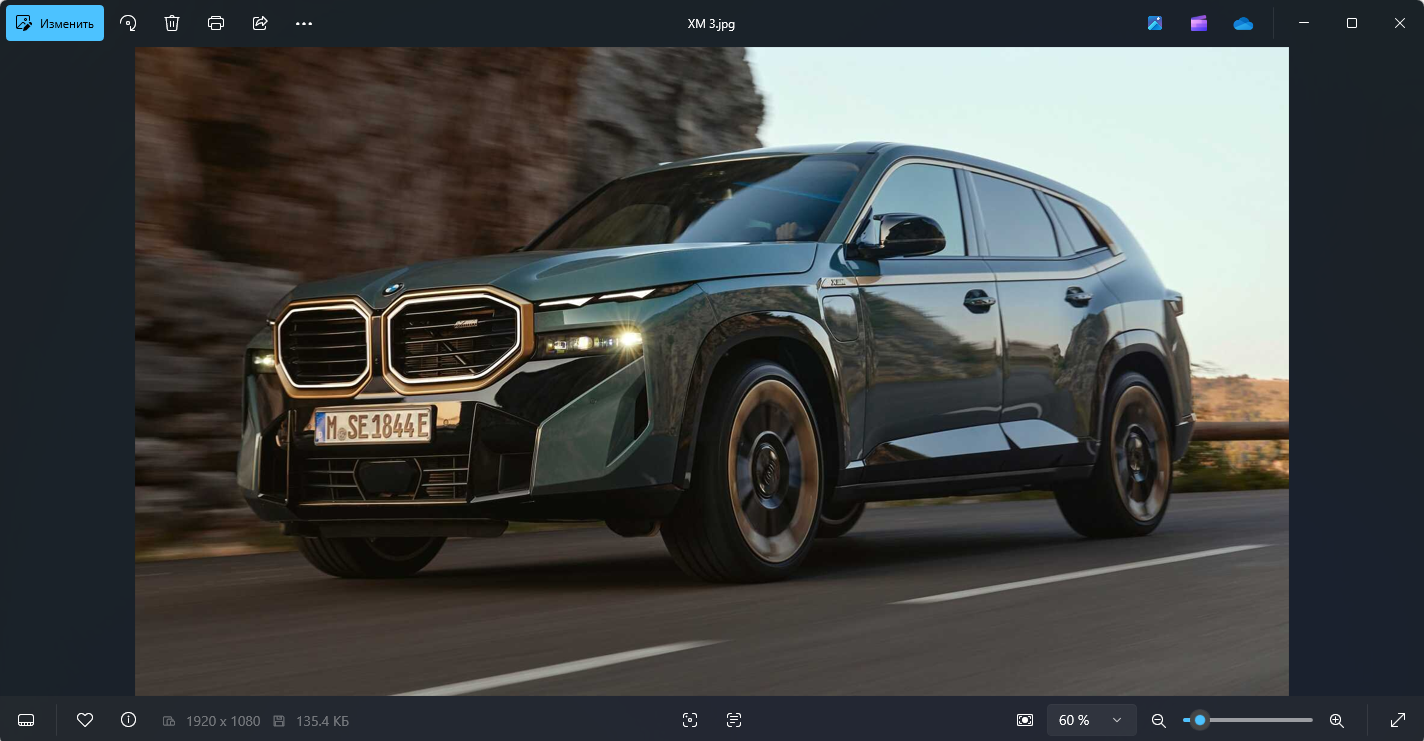


Расшифрованный текст:

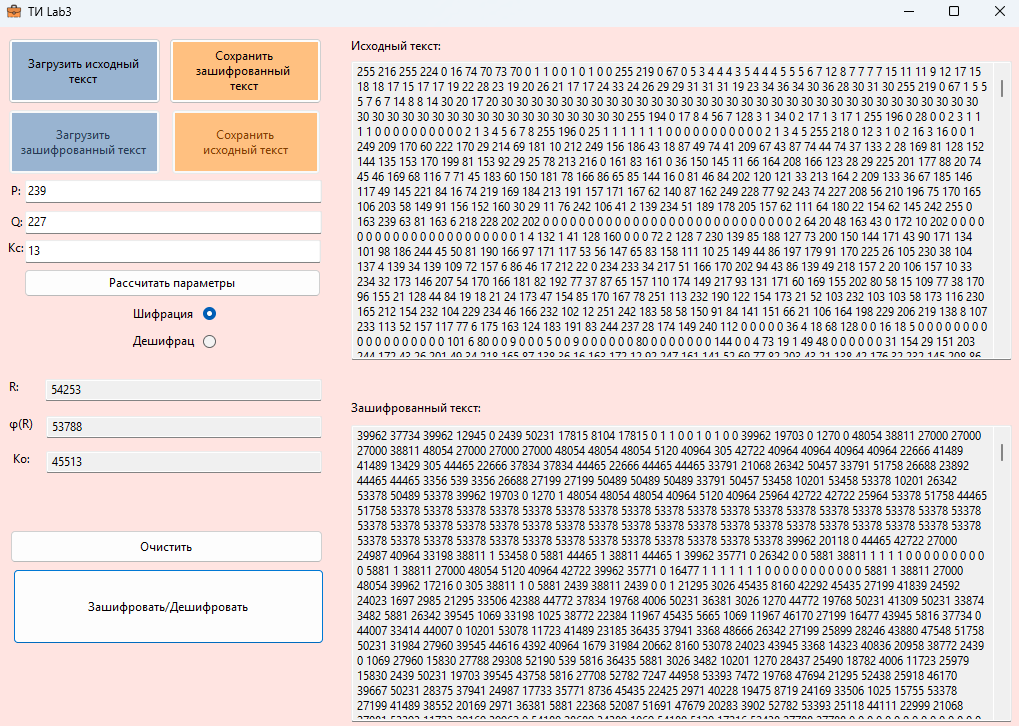


1. Шифрование фото:

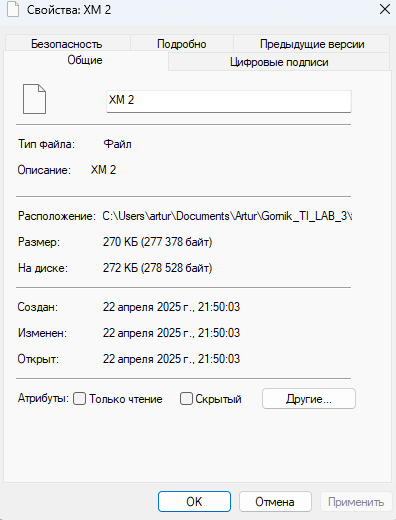


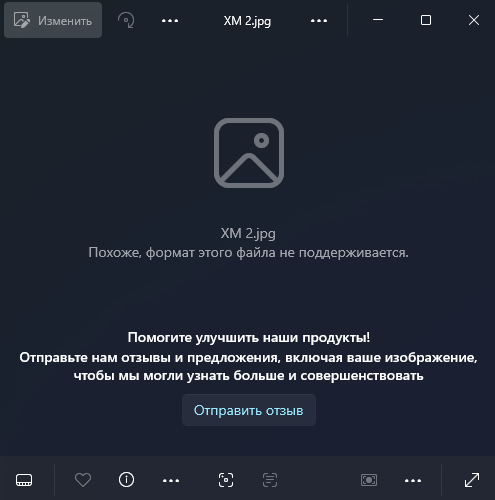


Результат работы шифратора:



Зашифрованная фотография:





Результат работы дешифратора:

