УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе № 3

по предмету «Теория информации»

Вариант 1

Выполнил:

Гузаев Е.Д

Гр. 351003

Проверил:

Болтак С. В.

Минск 2025

# Примеры алгоритмов для криптографии RSA

## 1. Алгоритм быстрого возведения в степень (7^15 mod 19)

Давайте проверим вычисление 7^15 mod 19 по алгоритму быстрого возведения в степень:

| **Шаг** | **a** | **n** | **e** | **Действие** | **r mod x** |
| --- | --- | --- | --- | --- | --- |
| 1 | 7 | 15 | нет | r = 1 \* 7 mod 19 = 7 n = 15 - 1 = 14 | 7 |
| 2 | 7 | 14 | да | a = 7² mod 19 = 49 mod 19 = 11 n = 14 / 2 = 7 | 7 |
| 3 | 11 | 7 | нет | r = 7 \* 11 mod 19 = 77 mod 19 = 1 n = 7 - 1 = 6 | 1 |
| 4 | 11 | 6 | да | a = 11² mod 19 = 121 mod 19 = 7 n = 6 / 2 = 3 | 1 |
| 5 | 7 | 3 | нет | r = 1 \* 7 mod 19 = 7 n = 3 - 1 = 2 | 7 |
| 6 | 7 | 2 | да | a = 7² mod 19 = 49 mod 19 = 11 n = 2 / 2 = 1 | 7 |
| 7 | 11 | 1 | нет | r = 7 \* 11 mod 19 = 77 mod 19 = 1 n = 1 - 1 = 0 | 1 |

Результат: 7^15 mod 19 = 1

## 2. Поиск первообразных корней по модулю 17

Для простого числа p = 17 найдем все первообразные корни:

1. Находим разложение φ(17) = p-1 = 16 = 2⁴
2. Проверим, является ли число 3 первообразным корнем:
   * 3^(16/2) = 3^8 mod 17 = 6561 mod 17 = 16 ≠ 1
   * Поскольку 3^8 ≠ 1 (mod 17), число 3 является первообразным корнем
3. Генерация остальных первообразных корней:
   * Если g является первообразным корнем, то g^k тоже первообразный корень, если gcd(k, p-1) = 1
   * Для p = 17 значения k, взаимно простые с 16: 1, 3, 5, 7, 9, 11, 13, 15
   * Рассчитаем все первообразные корни:
     + 3^1 mod 17 = 3
     + 3^3 mod 17 = 27 mod 17 = 10
     + 3^5 mod 17 = 243 mod 17 = 5
     + 3^7 mod 17 = 2187 mod 17 = 11
     + 3^9 mod 17 = 19683 mod 17 = 16
     + 3^11 mod 17 = 177147 mod 17 = 14
     + 3^13 mod 17 = 1594323 mod 17 = 8
     + 3^15 mod 17 = 14348907 mod 17 = 7

Таким образом, все первообразные корни по модулю 17 это: 3, 5, 7, 8, 10, 11, 14, 16

## 3. Расширенный алгоритм Евклида для взаимно простых чисел

Для нахождения xa + yb = НОД(a,b), где a = 841, b = 299:

## Таблица вычисления НОД:

| **Итерация** | **Делимое** | **Делитель** | **Частное** | **Остаток** |
| --- | --- | --- | --- | --- |
| 1 | 841 | 299 | 2 | 243 |
| 2 | 299 | 243 | 1 | 56 |
| 3 | 243 | 56 | 4 | 19 |
| 4 | 56 | 19 | 2 | 18 |
| 5 | 19 | 18 | 1 | 1 |
| 6 | 18 | 1 | 18 | 0 |

НОД(841, 299) = 1, что подтверждает взаимную простоту чисел.

## Обратный ход для нахождения x и y:

1. Из предпоследней строки: 1 = 19 - 18\*1
2. Подставляем выражение для 18 из строки выше: 18 = 56 - 1921 = 19 - (56 - 192)1 = 19 - 56 + 192  
   1 = 19\*3 - 56
3. Подставляем выражение для 19 из строки выше: 19 = 243 - 5641 = (243 - 564)3 - 561 = 2433 - 5612 - 561 = 2433 - 56\*13
4. Подставляем выражение для 56 из строки выше: 56 = 299 - 24311 = 2433 - (299 - 2431)131 = 2433 - 29913 + 243131 = 24316 - 299\*13
5. Подставляем выражение для 243 из первой строки: 243 = 841 - 29921 = (841 - 299*2)*16 - 299*13  
   1 = 841*16 - 299*32 - 299*131 = 841*16 - 299\*45*

Получаем решение: x = 16, y = -45

Проверка: 16\*841 + (-45)\*299 = 13456 - 13455 = 1

Таким образом, для уравнения x841 + y299 = 1, коэффициенты x = 16, y = -45.

**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.

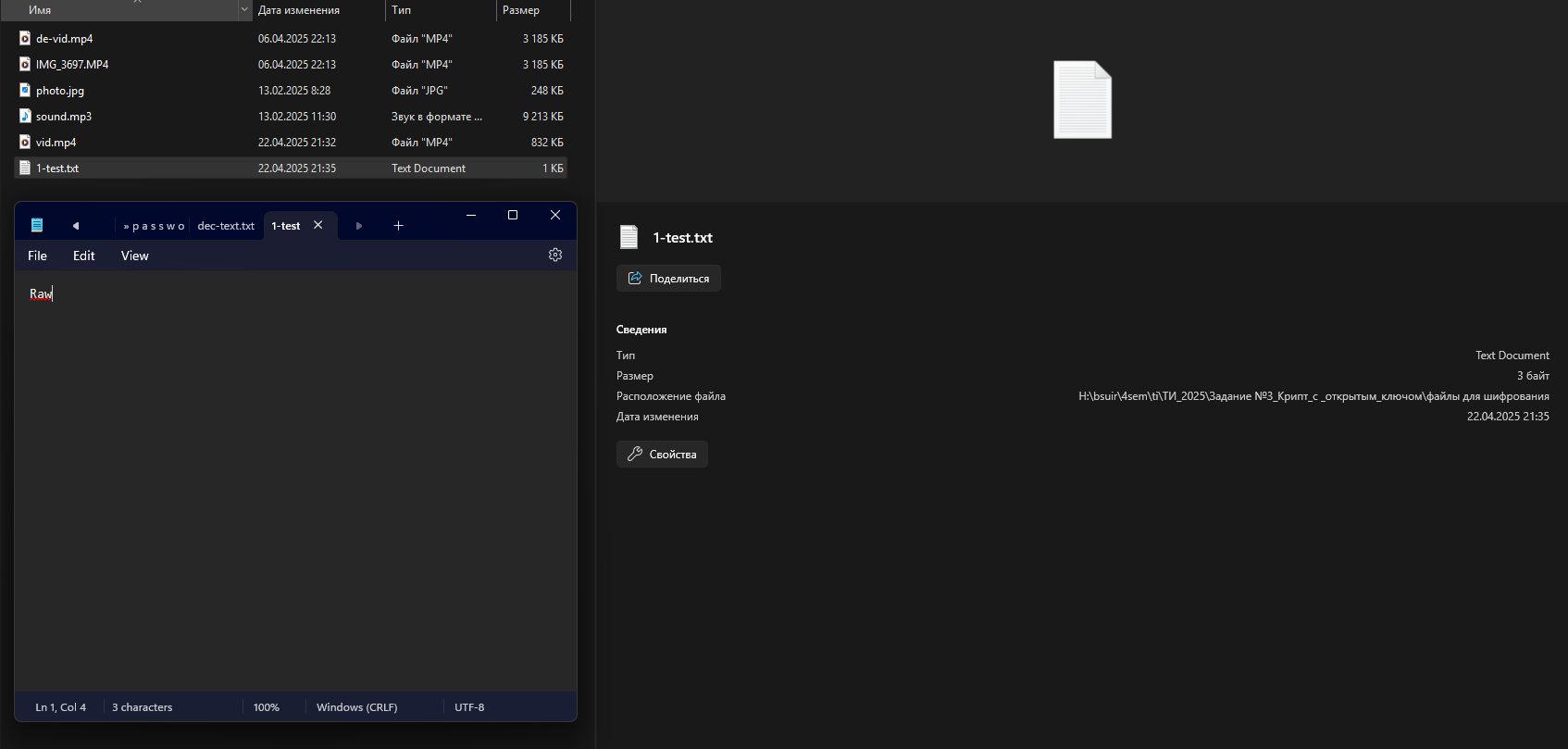
При использовании длинной арифметики для определения простоты числа использовать один из вероятностных тестов: тест Ферма или тест Миллера-Рабина.

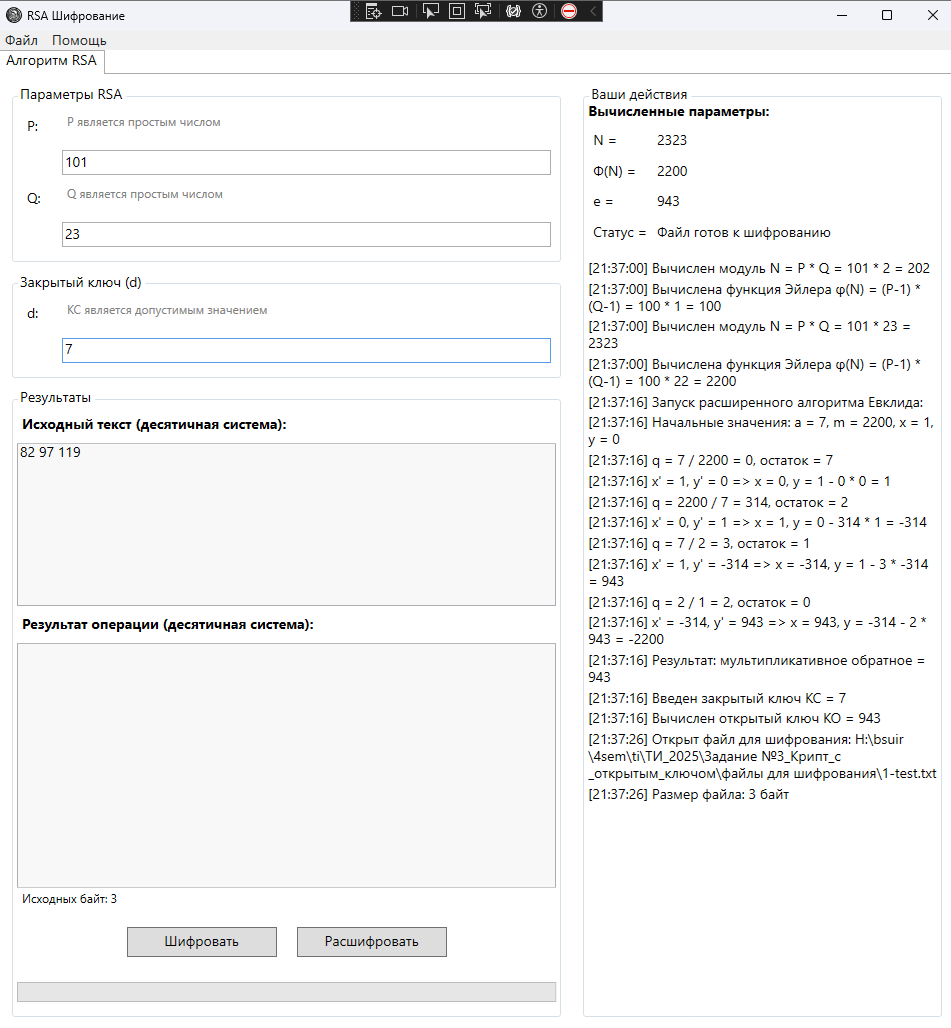
Результат работы программы – зашифрованный/расшифрованный файл/ы.

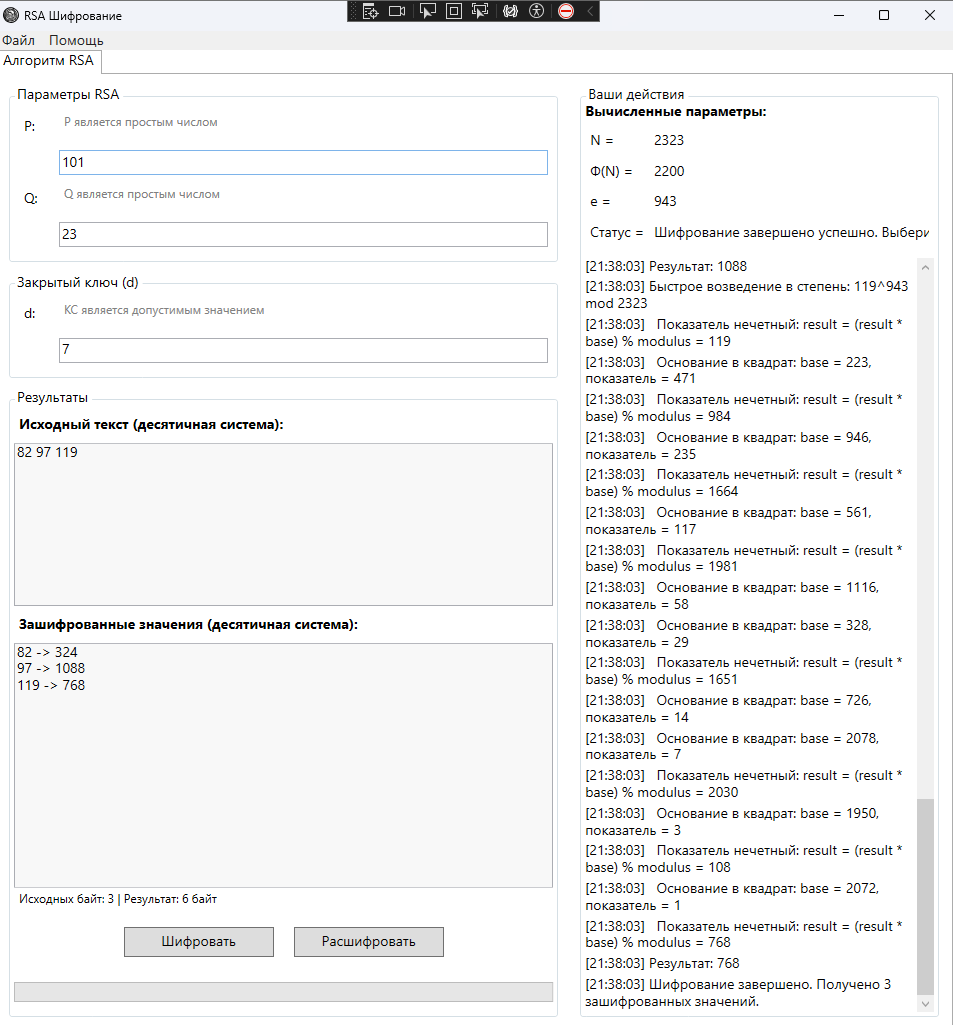
**Тесты на разные типы файлов**

1. Текстовый файл:

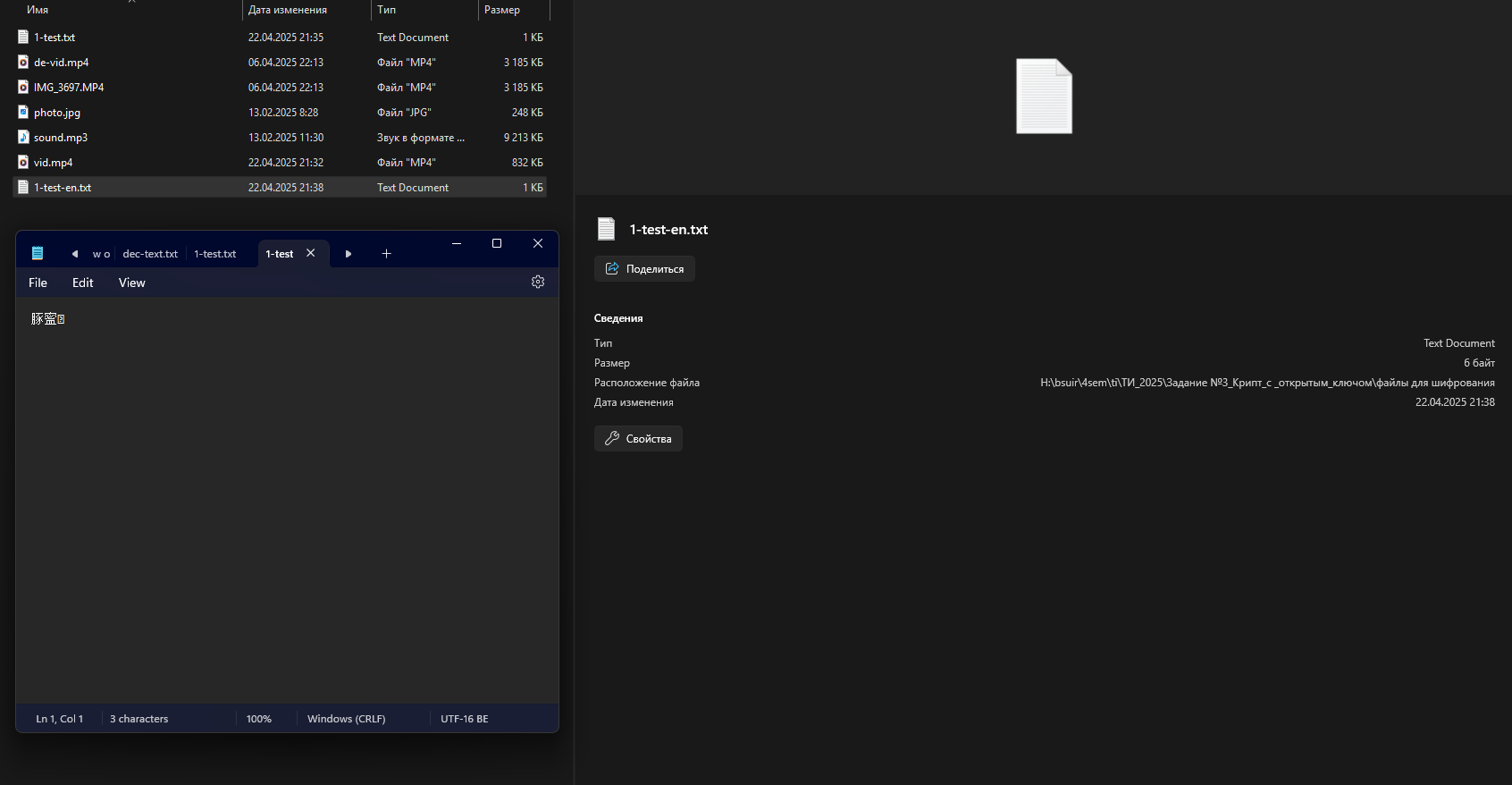
Исходное содержимое:

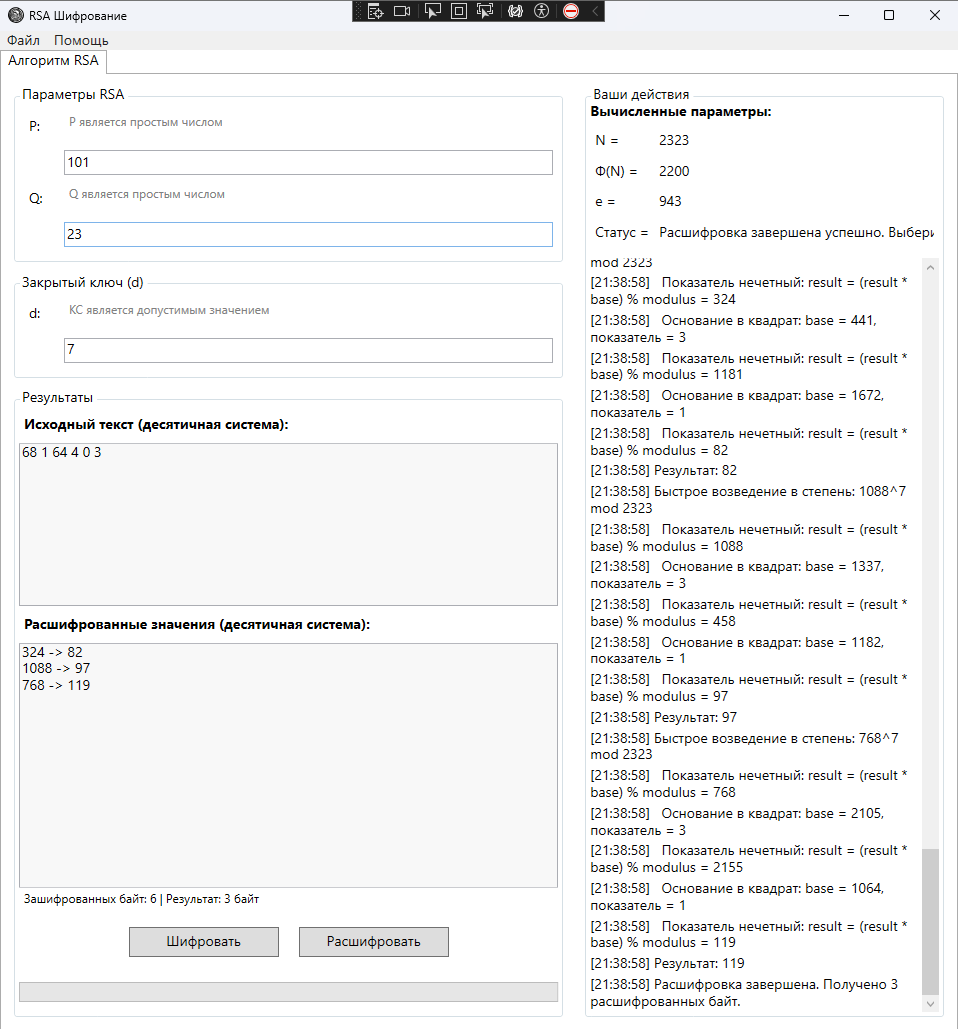


Отображение в программе:  


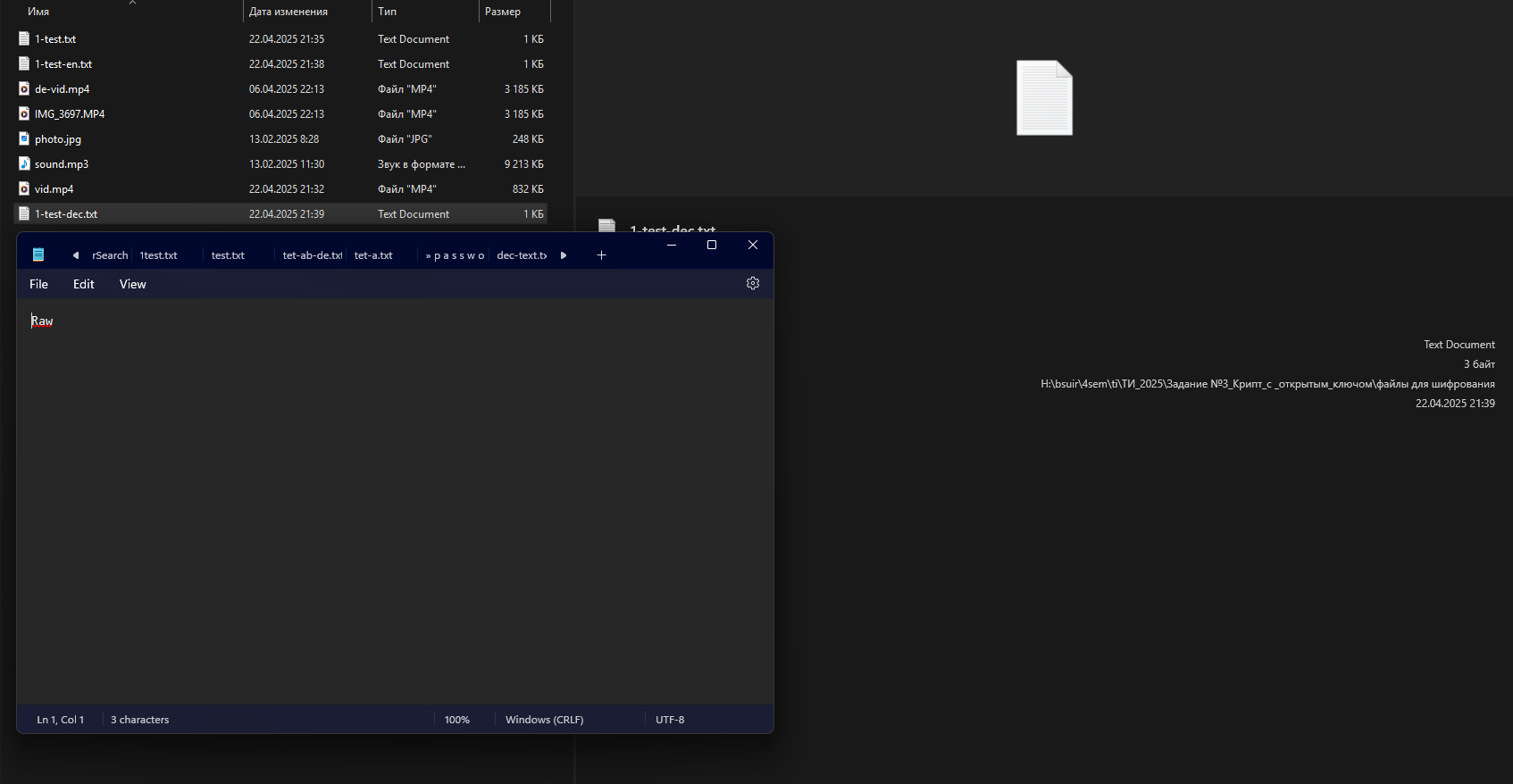
Результат шифрования:  


Отображение зашифрованного содержимого в файле:



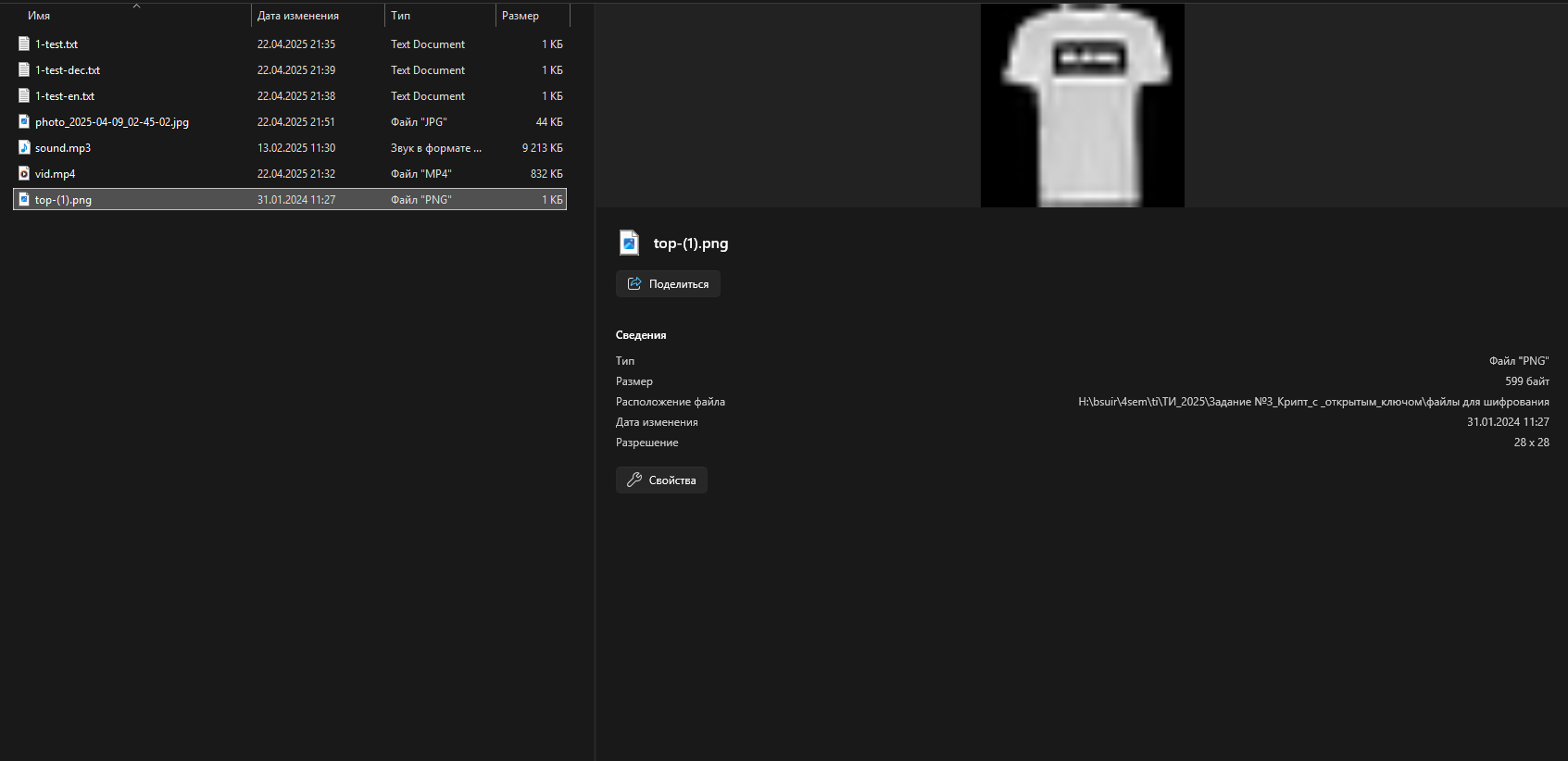
Результат расшифровки:  


Результат расшифровки записанный в файл:

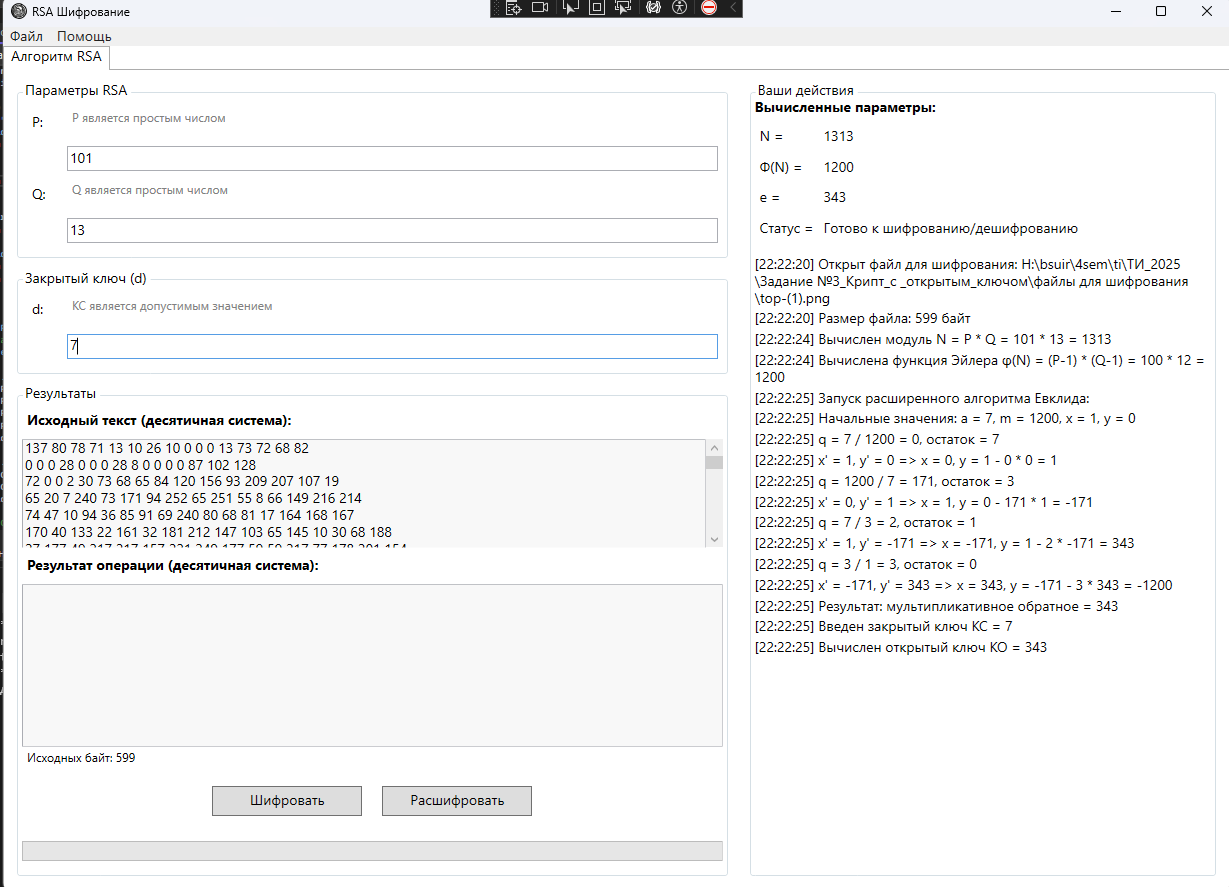


1. Изображение(файл):

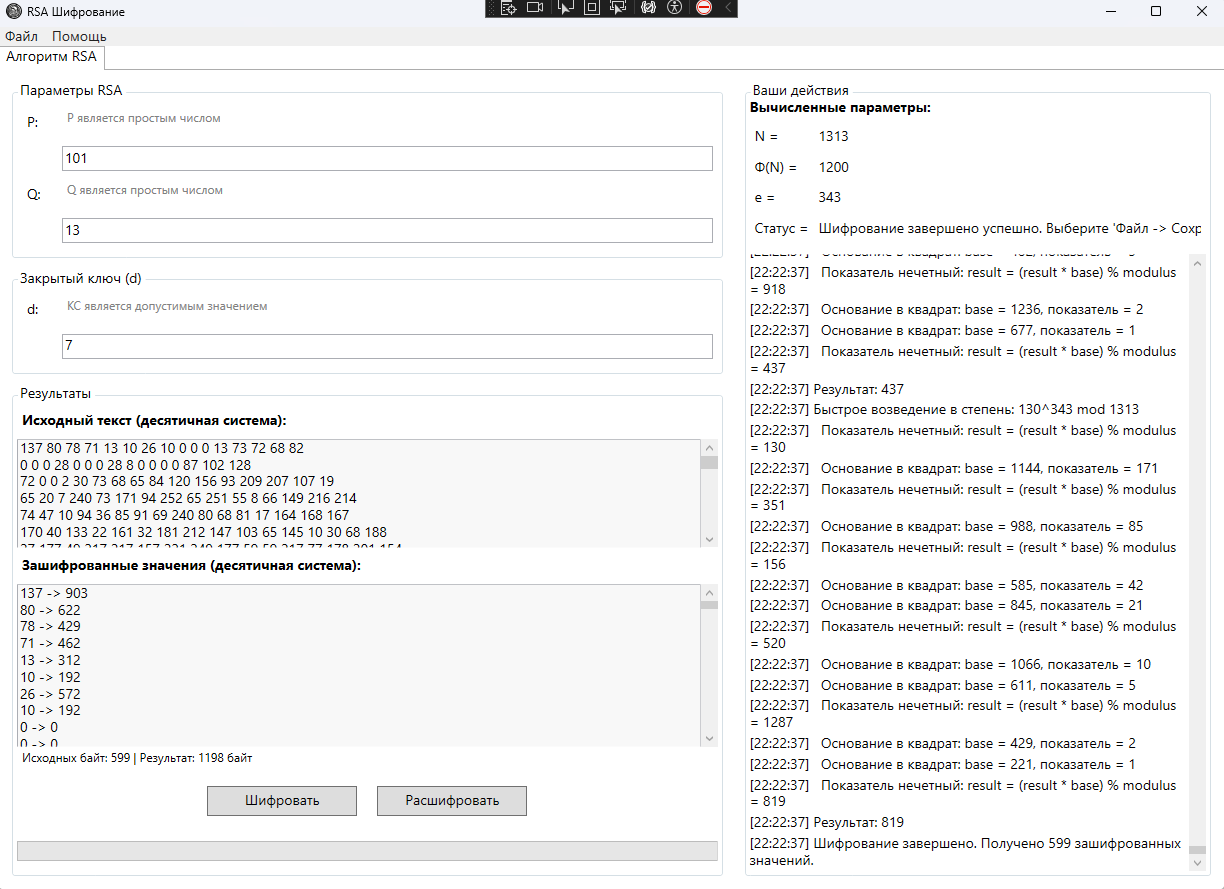
Исходное содержимое:



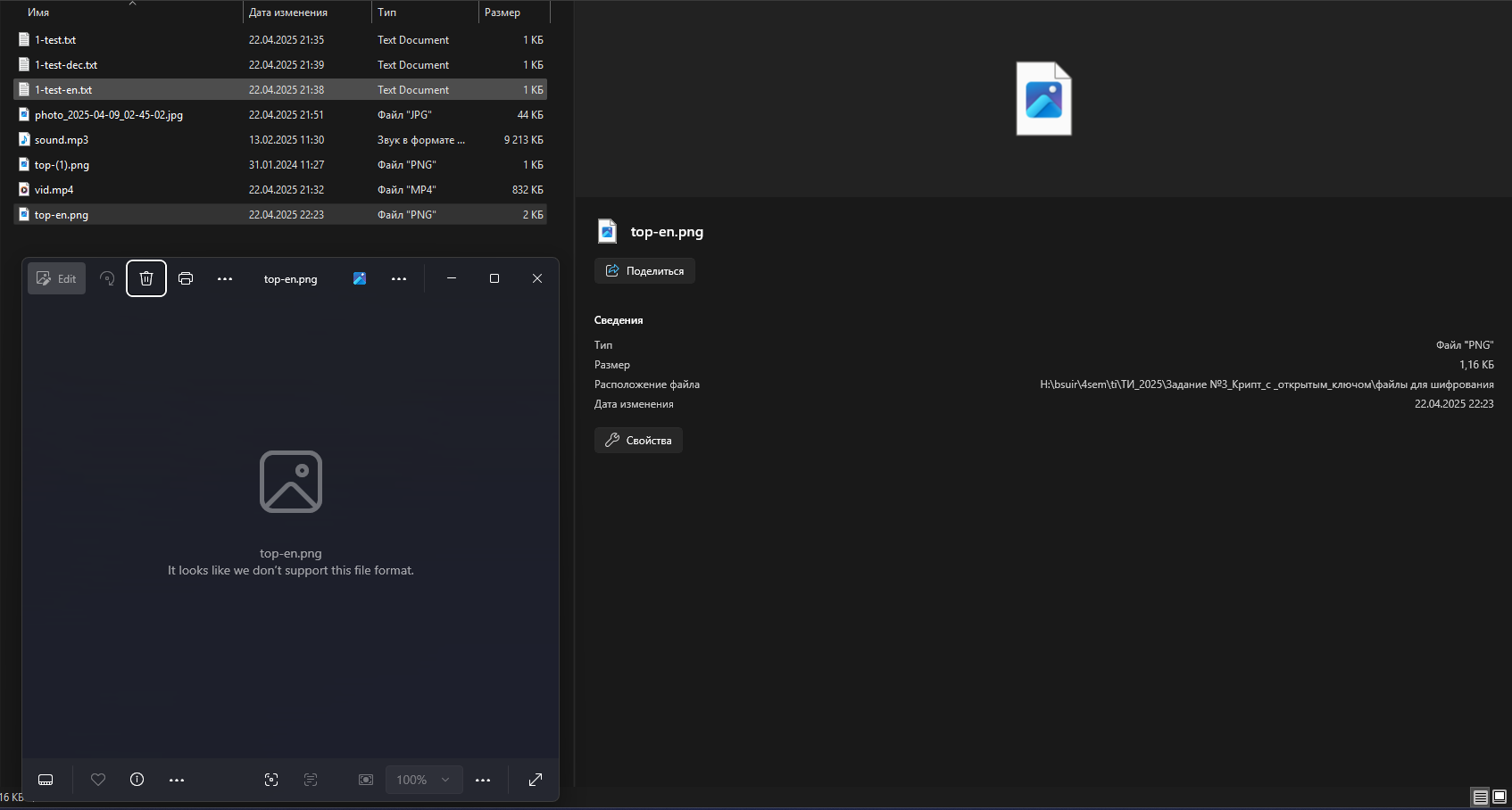
Отображение в программе:

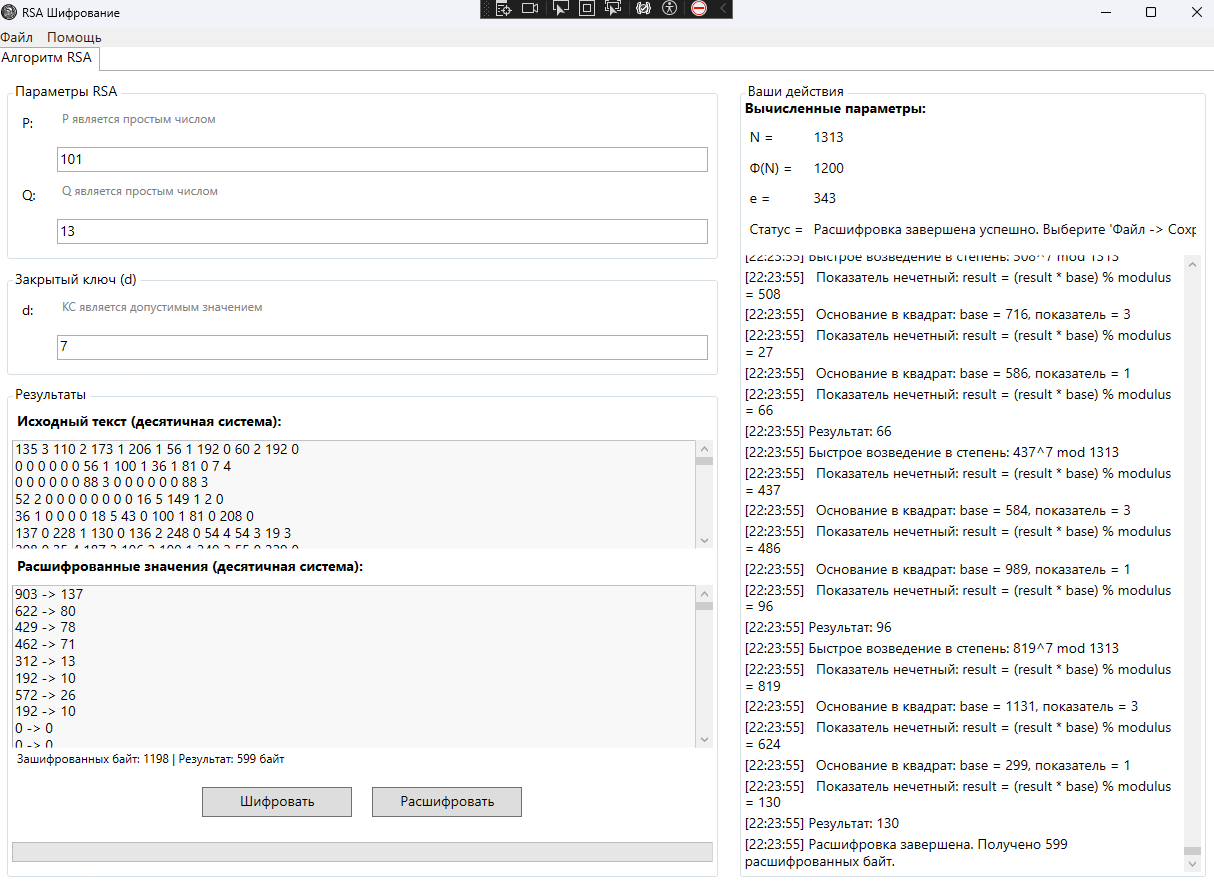


Результат шифрования:



Отображение зашифрованного содержимого в файле:



Результат расшифровки:  


Результат расшифровки записанный в файл:  
