**Orynbassar Nurlybek, 22B030575, 16:00-17:00 Thursday**

**Задание по прикладной криптографии: Программная реализация RSA с графическим интерфейсом**

**Цель:** Разработать программное приложение, реализующее алгоритм RSA для шифрования и расшифрования сообщений, с графическим интерфейсом.

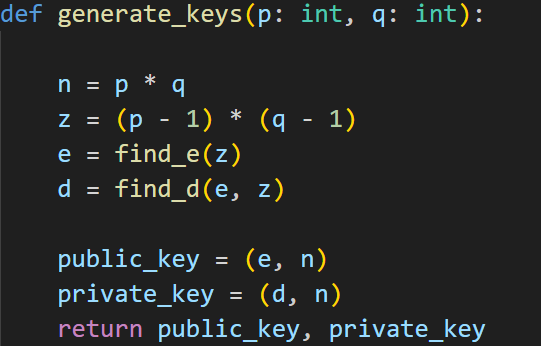
**1. Введение**

В данном проекте реализован алгоритм RSA для шифрования и расшифрования сообщений. Код включает в себя генерацию ключей, процесс шифрования и расшифрования, а также вспомогательные функции для нахождения параметров алгоритма. В дополнение к этому реализован графический интерфейс (GUI) для удобного взаимодействия с алгоритмом.

**2. Генерация ключей**

Функция generate\_keys(p, q) выполняет следующие действия:

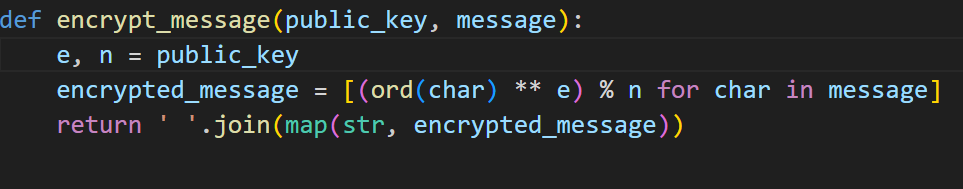
* Вычисляет произведение двух простых чисел .
* Определяет значение функции Эйлера .
* Подбирает значение открытой экспоненты e с помощью функции find\_e(z).
* Вычисляет секретную экспоненту d с помощью функции find\_d(e, z).
* Возвращает пару ключей: открытый (e, n) и закрытый (d, n).



**3. Процесс шифрования**

Функция encrypt\_message(public\_key, message) выполняет шифрование переданного сообщения следующим образом:

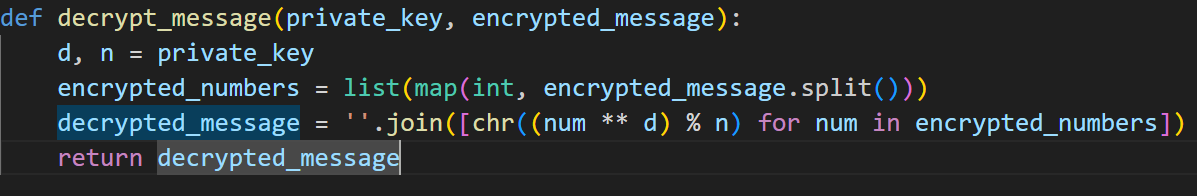
* Берёт открытый ключ (e, n).
* Преобразует каждый символ сообщения в его ASCII-код.
* Использует формулу для получения зашифрованного представления.
* Возвращает зашифрованное сообщение в виде строки из чисел.



**4. Процесс расшифрования**

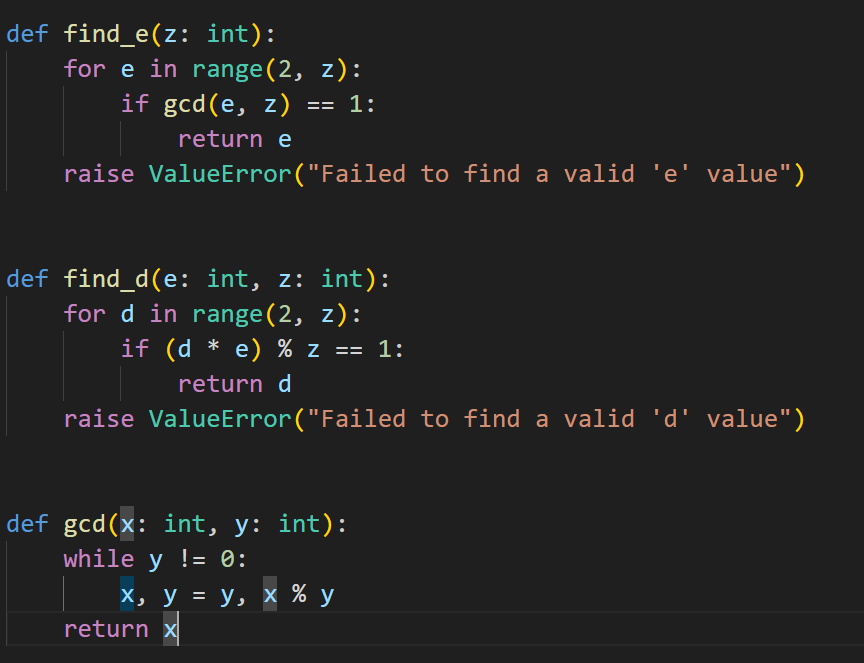
Функция decrypt\_message(private\_key, encrypted\_message) выполняет обратный процесс:

* Использует закрытый ключ (d, n).
* Разбивает строку зашифрованных чисел на отдельные элементы.
* Применяет формулу для восстановления исходного текста.
* Возвращает расшифрованное сообщение.



**5. Вычисление параметров**

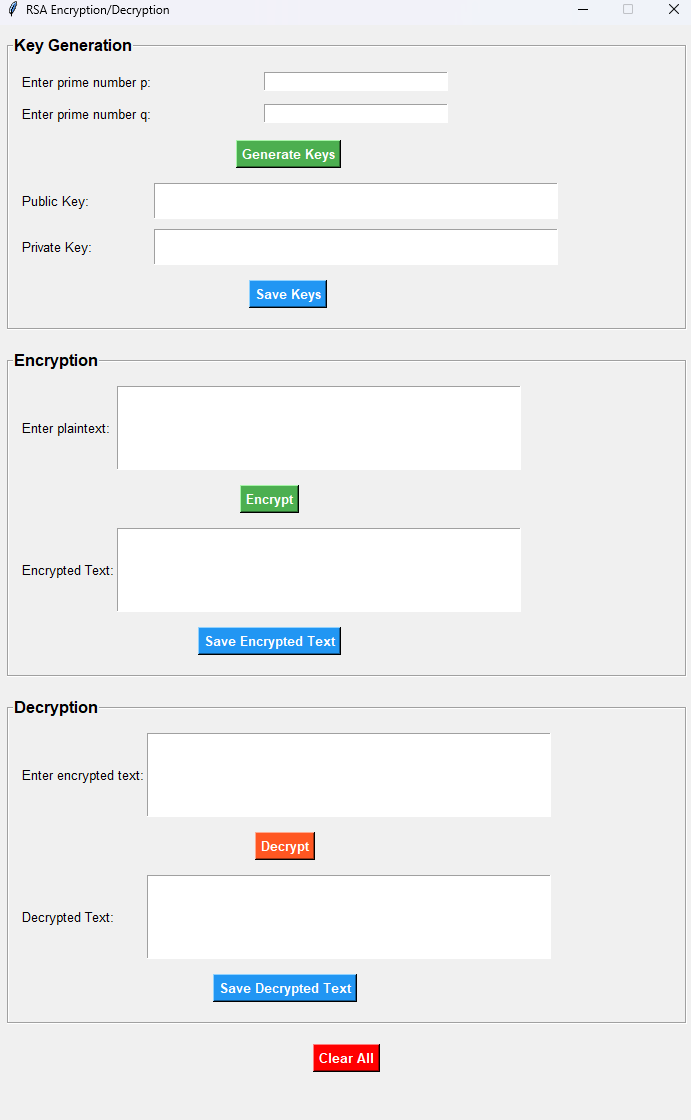
* find\_e(z): Подбирает значение e, которое является взаимно простым с z.
* find\_d(e, z): Подбирает значение d, которое удовлетворяет уравнению .
* gcd(x, y): Вычисляет наибольший общий делитель двух чисел с помощью алгоритма Евклида.



**6. Графический интерфейс**

Для удобного взаимодействия с алгоритмом был разработан GUI на базе Tkinter. Интерфейс включает в себя:

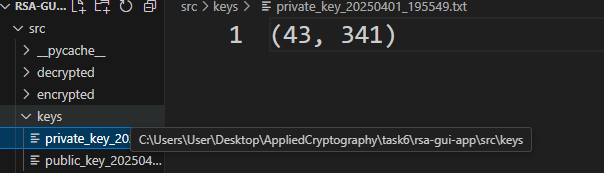
* Поля ввода для простых чисел p и q.
* Кнопку "Generate Keys" для генерации ключей.
* Поля вывода открытого и закрытого ключей.
* Кнопки для сохранения ключей в файлы.
* Поле ввода открытого текста и кнопку "Encrypt" для шифрования.
* Поле вывода зашифрованного текста и кнопку "Save Encrypted Text" для сохранения.
* Поле ввода зашифрованного текста и кнопку "Decrypt" для расшифрования.
* Поле вывода расшифрованного текста и кнопку "Save Decrypted Text".
* Кнопку "Clear All" для очистки всех полей.



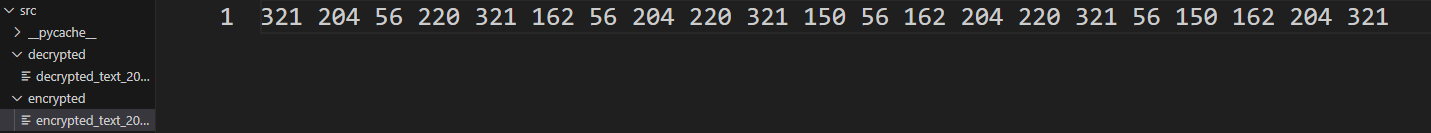
**7. Работа с файлами**

Приложение позволяет сохранять и загружать ключи, зашифрованные и расшифрованные сообщения:

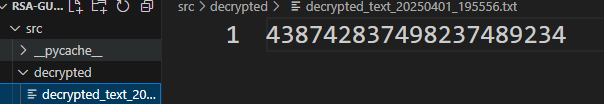
* Ключи сохраняются в папку keys/ с временной меткой.



* Зашифрованные сообщения сохраняются в encrypted/.



* Расшифрованные сообщения сохраняются в decrypted/.



**8. Заключение**

Реализованный алгоритм позволяет выполнять базовые операции RSA-шифрования через удобный графический интерфейс. Для дальнейшего развития можно добавить:

* Генерацию случайных простых чисел p и q.
* Улучшение производительности с помощью модульной арифметики.
* Возможность выбора размера ключей (1024, 2048, 4096 бит).
* Поддержку работы с текстовыми файлами для шифрования и дешифрования.