Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ОТЧЕТ

О ЛАБОРАТОРНОЙ РАБОТЕ

Лабораторная работа №4

по теме:

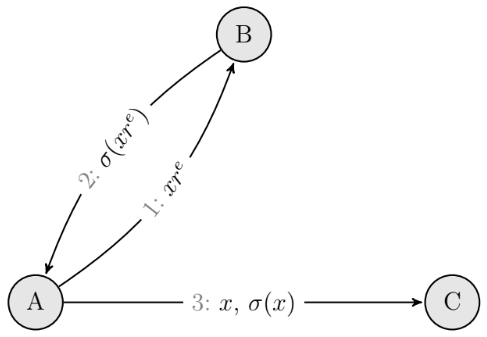
«Создание и верификация цифровой подписи»

Исполнитель, студент группы 201-361 \_\_\_\_\_\_\_\_\_\_\_\_ Н.А. Фельдбуш

Москва, 2023

**Постановка задачи**:

Реализуйте простое клиент-серверное приложение, позволяющее аккумулировать короткие анонимные сообщения (систему электронного голосования) согласно следующей схеме:



Здесь: A – пользователь (избиратель), B – регистратор, C – счетчик, x – сообщение (голос), r – известное только участнику A случайное число, (e, n) – открытый ключ банка. Пренебрегите реализацией правильных механизмов распределения, хранения и сертификации ключей.

**Ход работы:**

Выполним поставленную задачу при помощи средств языка программирования Python. Перед началом следует обратить внимание на порядок операций, выполняемых пользователем, регистратором и счетчиком. В первую очередь регистратор формирует у себя закрытый ключ, по которому формирует открытый который позже передает пользователю. Затем пользователь с помощью модуля открытого ключа формирует случайное число r. После этого используя модуль экспоненту и число r ослепляет сообщение и отправляет его регистратору, который в свою очередь подписывает его с помощью модуля и экспоненты закрытого ключа, а затем отправляет обратно пользователю. Пользователь проводит операцию обратную операции ослепления и получает подписанный закрытым ключом исходный файл, который затем в комбинации с изначальным файлом отправляется счетчику. Счетчик с помощью функции подтверждения подписи проверяет, действительна ли она или нет.

Итак:

1. Регистратор формирует закрытый и открытый ключ (рис. 1-2):



Рисунок 1 – Закрытый ключ



Рисунок 2 – Открытый ключ

1. Передача открытого ключа пользователю (рис. 3):

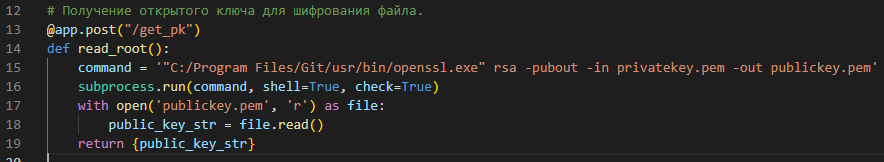


Рисунок 3 – Передача ключа

1. Получение числа r (рис. 4):

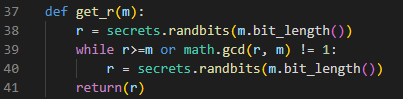


Рисунок 4 – Число r

1. Ослепление сообщения (рис. 5):



Рисунок 5 – Функция ослепления

1. Отправка сообщения (рис. 6):



Рисунок 6 – Отправка сообщения

1. Подписание с помощью модуля и экспоненты закрытого ключа (рис. 7):

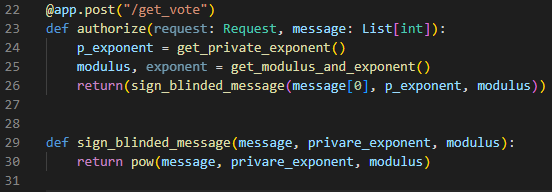


Рисунок 7 – Подписание с помощью модуля и экспоненты

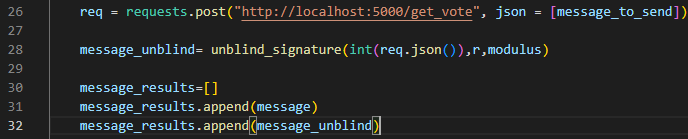
1. Получение подписанного сообщения клиентом и операция обратная ослеплению (рис. 8):

Рисунок 8 – Операция обратная ослеплению

1. Отправка файлов счетчику (рис. 9):

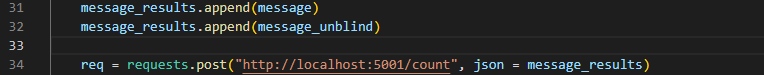


Рисунок 9 – Отправка изначального файла и подписанного

1. Функция подтверждения у счетчика (рис. 10):



Рисунок 10 – Функция подтверждения

Проверим работу приложения. Зададим количество избирателей равным трем участникам, запустим все три класса в необходимом порядке и рассмотрим содержимое консоли (рис.14). Консоль разделена на три части: слева – ClientA, по середине – ServerB, справа – ServerC. Можем заметить, как клиент отправляет замаскированное сообщение серверу B, сервер B его обрабатывает, а затем сервер C получает голос (рис. 11).

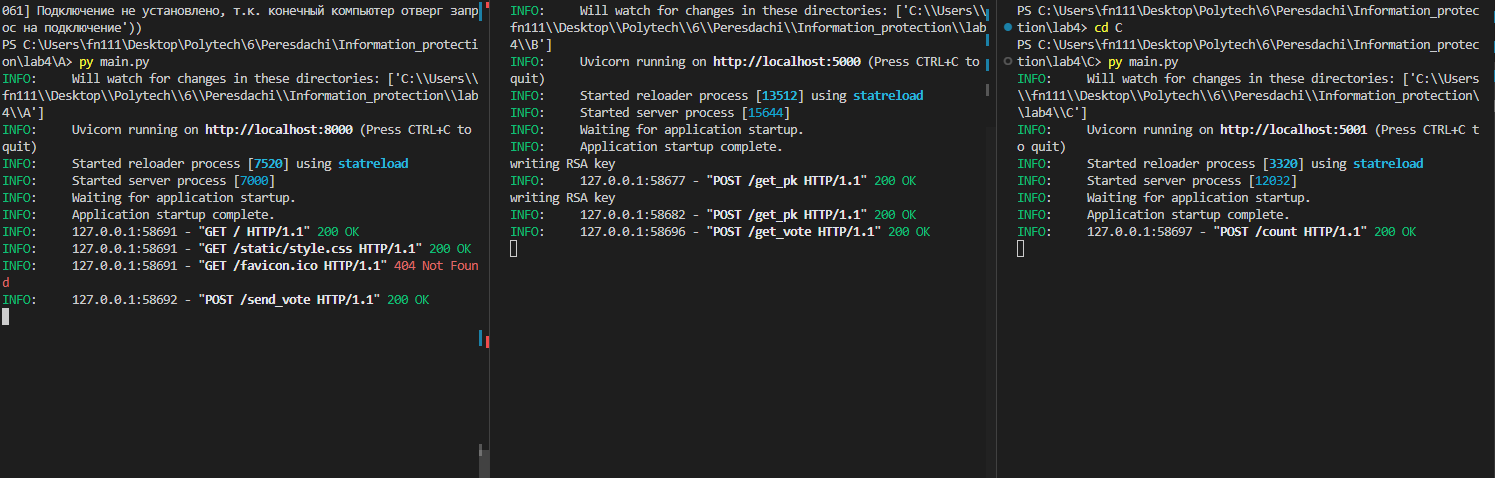


Рисунок 11 – Работа приложения

Также дополнительно визуализирован подсчет голосов (рис. 12):

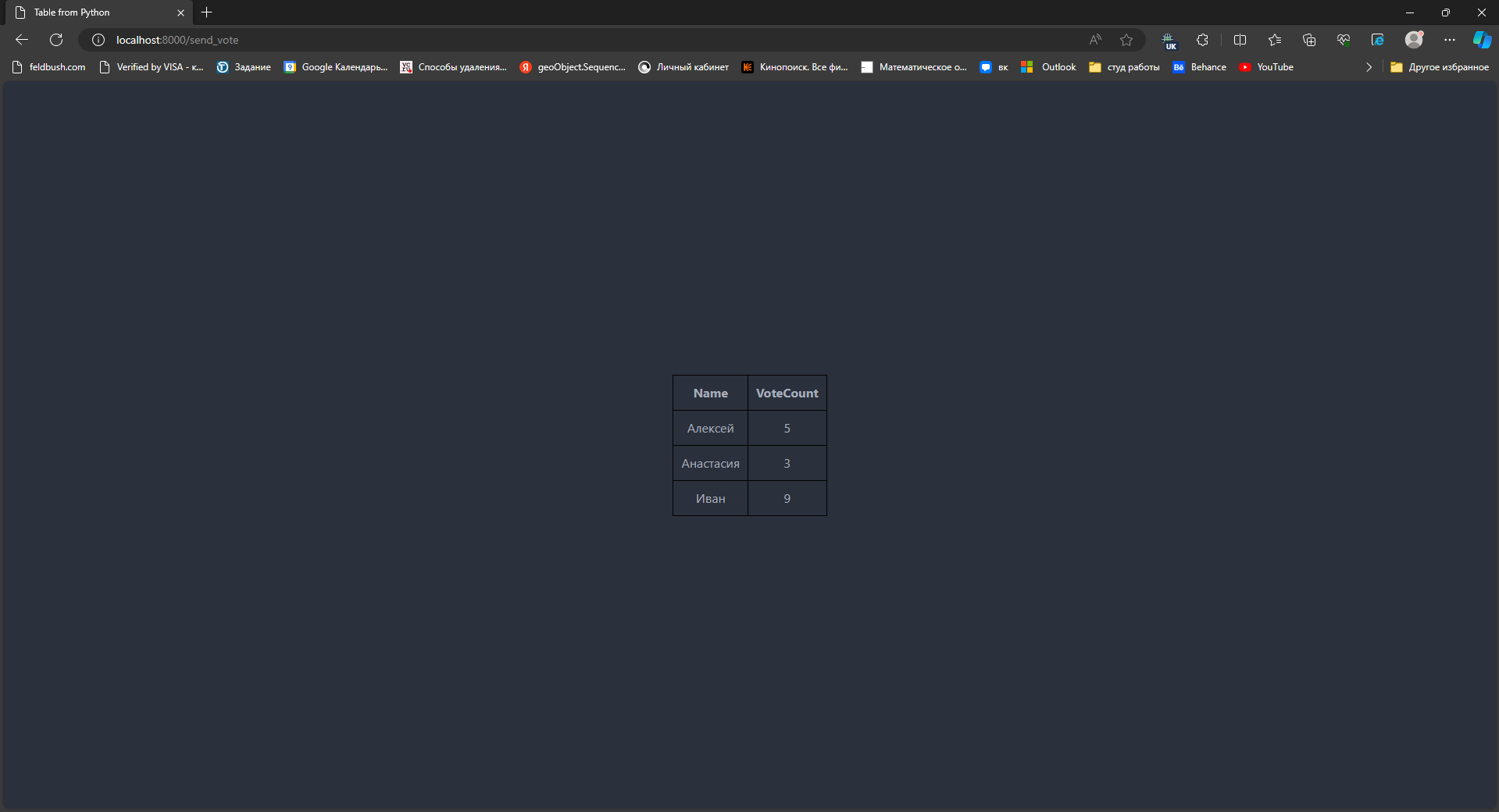


Рисунок 12 – Работа приложения