**Алгоритмы асимметричного шифрования**

**Цель работы**

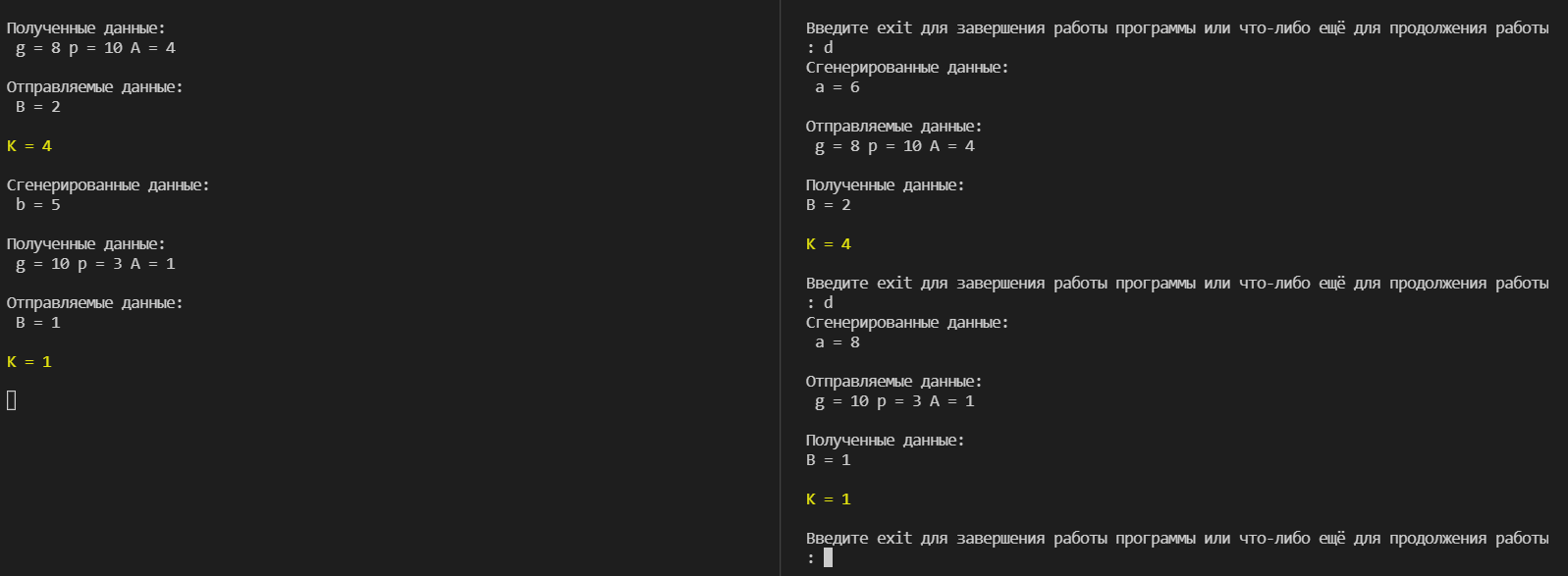
Познакомиться с принципами работы протоколов рукопожатия в современных компьютерных системах

**Задания для выполнения**

1. Реализовать протокол Диффи-Хеллмана в виде клиент-серверного приложения.

Реализация: файл server1.py и client1.py

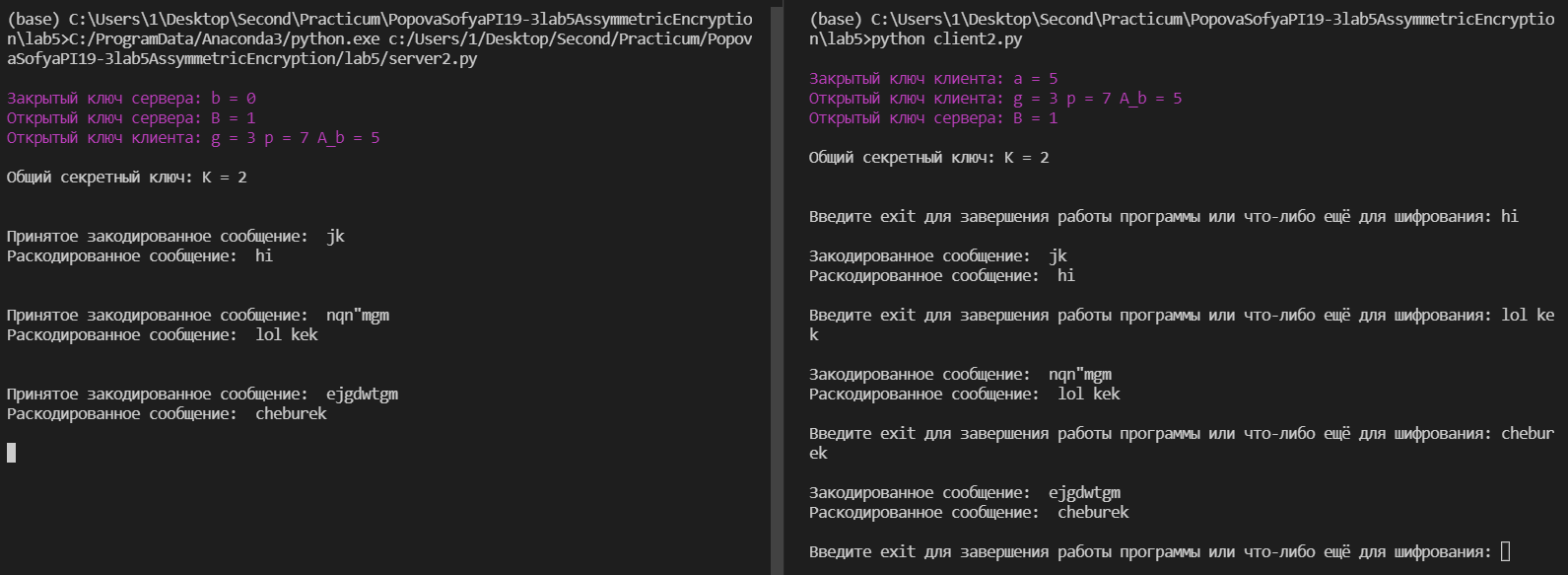
Скрин работы:



При любых рандомно генерируемых данных ответ K получается один и тот же.

2. Реализовать клиент-серверную пару, которая шифрует сообщения асимметричным способом.

Программа шифрует сообщения на основе общего секретного ключа, файлы server2.py и client2.py. Пример работы:

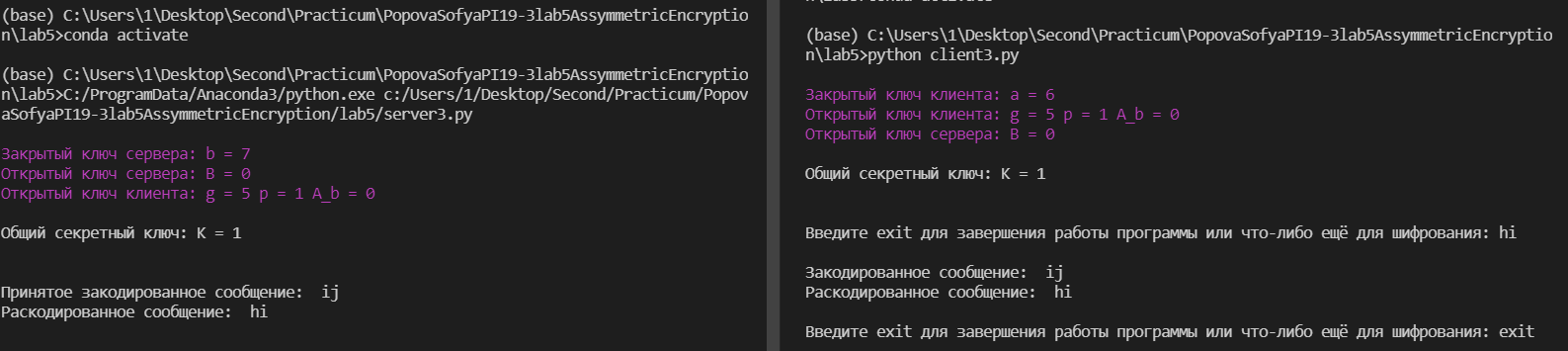


**Дополнительные задания**

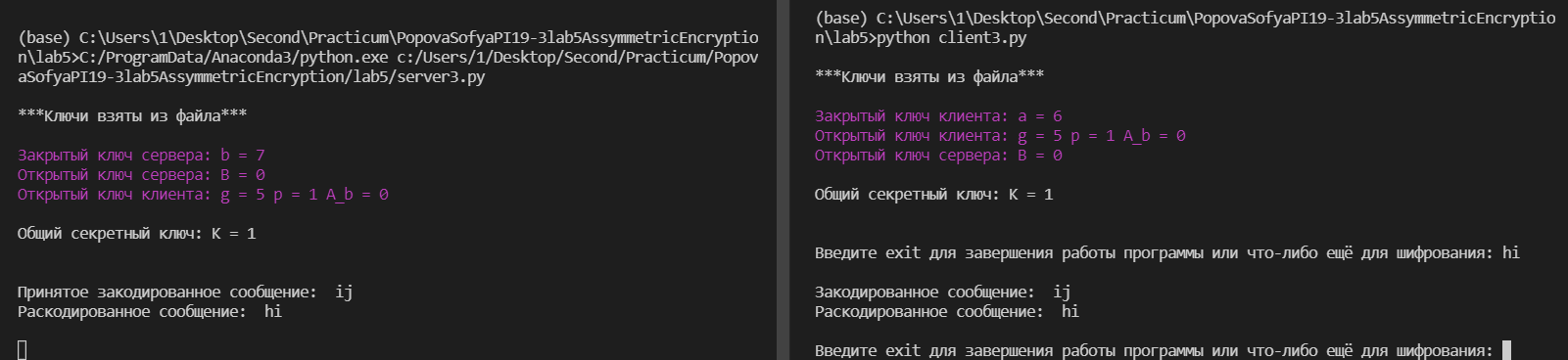
Реализованы в server3.py и client3.py

1. Модифицируйте код клиента и сервера так, чтобы приватный и публичный ключ хранились в текстовых файлах на диске и, таким образом, переиспользовались между запусками.

Первый запуск с генерацией данный (когда файла настроек ещё не существует):



Запуск с уже существующими настройками:

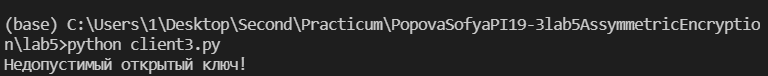


2. Проведите рефакторинг кода клиента и сервера так, чтобы все, относящееся к генерации ключей, установлению режима шифрования, шифрованию исходящих и дешифрованию входящих сообщений было отделено от основного алгоритма обмена сообщениями.

Оно отделено.

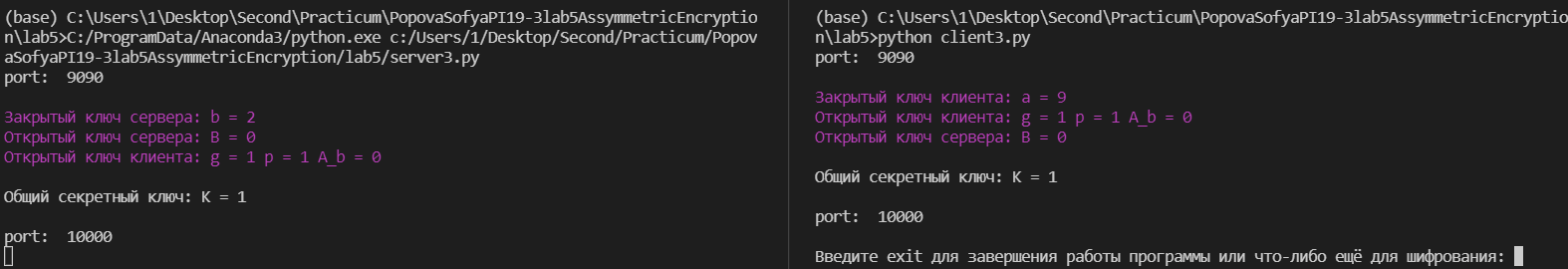
3. Реализуйте на сервере проверку входящих сертификатов. На сервере должен храниться список разрешенных ключей. Когда клиент посылает на сервер свой публичный ключ, сервер ищет его среди разрешенных и, если такого не находит, разрывает соединение. Проверьте правильность работы не нескольких разных клиентах.

Вообще у меня происходит генерация данных, но если кто-то решит изменить данные на недопустимые, в ответ выйдет ошибка:



4. Модифицируйте код клиента и сервера таким образом, чтобы установление режима шифрования происходило при подключении на один порт, а основное общение - на другом порту. Номер порта можно передавать как первое зашифрованное сообщение.

Порт меняется:



5. Реализуйте пул портов.

Добавила.

6. Модифицируйте код FTP-сервера таким образом, чтобы он поддерживал шифрование.

Поддерживает, у меня шифруются данные для авторизации/регистрации аналогичным образом:

