|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **МИНОБРНАУКИ РОССИИ**  федеральное государственное бюджетное образовательное  учреждение высшего образования  **«Национальный исследовательский университет «МЭИ»** | | | | | **Институт** | | ИВТИ | | | **Кафедра** | | ПМИИ | | |  |  | |   **Отчет по курсовой работе**  **Дисциплина:**  **«Защита данных»**  **«Программная реализация протокола Диффи-Хеллмана»**  Выполнила: студент группы А-13а-20  Алибеков А.А.  Преподаватель: Хорев П.Б.  Москва  2023 |

**Оглавление**

[**1. Введение** 3](#_Toc149310727)

[**1)** **Цель работы** 3](#_Toc149310728)

[**2)** **Задачи** 3](#_Toc149310729)

[**3)** **Описание протокола Диффи-Хеллмана** 3](#_Toc149310730)

[**4)** **Описание алгоритма** 3](#_Toc149310731)

[**2. Результаты проектирования программы.** 6](#_Toc149310732)

[**Интерфейс** 6](#_Toc149310733)

[**3. Тестирование разработанной программы.** 8](#_Toc149310734)

[**Заключение** 20](#_Toc149310735)

[**Список литературы** 20](#_Toc149310736)

[**Приложение** 20](#_Toc149310737)

**1. Введение**

1. **Цель работы**

Программная реализация протокола Диффи-Хеллмана.

Создание защищенного сеанса связи между двумя(тремя) приложениями (или окнами одного приложения) на основе указанного протокола.

Возможность сохранения в выбираемом текстовом файле(журнале) записи выполнения реализованного протокола и просмотра этого журнала.

1. **Задачи**
2. Проектирование алгоритма решения протокола Диффи-Хеллмана.
3. Проектирование и реализация графического интерфейса.
4. Проектирование логики взаимодействия компонентов программы.
5. Реализация алгоритма протокола Диффи-Хеллмана для обмена ключами.
6. Реализация логирования действий алгоритма.
7. Тестирование и отладка программы.
8. **Описание протокола Диффи-Хеллмана**

Протокол Диффи-Хеллмана - криптографический протокол, позволяющий двум и более сторонам получить общий секретный ключ, несмотря на то, что их коммуникация происходит через незащищенный от прослушивания канал связи. Полученный ключ используется для шифрования дальнейшего обмена с помощью алгоритмов симметричного шифрования.

Инновационность протокола Диффи-Хеллмана заключается в решении одной из основных проблем классической криптографии, а именно, проблемы безопасного распределения ключа.

Важно отметить, что сам по себе алгоритм Диффи-Хеллмана уязвим к атакам, таким как модификация данных в канале связи и атаки “Man-in-the-middle”(атака человека посередине). Поэтому для обеспечения дополнительной безопасности при использовании этого протокола обычно используют дополнительные методы односторонней или двусторонней аутентификации. Эти методы помогают удостовериться в том, что обе стороны действительно общаются друг с другом и что нет нежелательных посредников в коммуникации.

1. **Описание алгоритма**

Предположим, существует два абонента: Алиса и Боб. Обоим абонентам известны некоторые два числа *g* и *p*, которые не являются секретными и могут быть известны также другим заинтересованным лицам. *g* обычно выбирается как первообразный корень по модулю *p*. Для того, чтобы создать неизвестный более никому секретный ключ, оба абонента генерируют большие случайные числа: Алиса — число *a*, Боб — число *b*.

Эти числа являются их закрытыми ключами и должны держаться в секрете. Затем Алиса вычисляет остаток от деления

(1)

будет публичным ключом Алисы.

и пересылает его Бобу, а Боб вычисляет остаток от деления

(2)

будет публичным ключом Боба.

и передаёт Алисе. Предполагается, что злоумышленник может получить оба этих значения, но не модифицировать их (то есть, у него нет возможности вмешаться в процесс передачи).

Теперь оба абонента могут вычислить общий секретный ключ *K*.

На втором этапе Алиса на основе имеющегося у неё *a* и полученного по сети *B* вычисляет значение:

(3)

Боб на основе имеющегося у него *b* и полученного по сети *A* вычисляет значение:

(4)

Как нетрудно видеть, у Алисы и Боба получилось одно и то же число

(5)

Его они и могут использовать в качестве секретного ключа, поскольку здесь злоумышленник встретится с практически неразрешимой (за разумное время) проблемой вычисления (3) или (4) по перехваченным и , если числа *p, a, b* выбраны достаточно большими.

Работа алгоритма показана на рисунке.

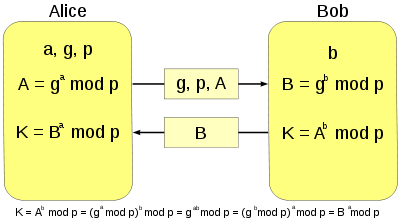


Рисунок 1.1 Работа алгоритма Диффи-Хеллмана

Алгоритм Диффи — Хеллмана, где K — итоговый общий секретный ключ

При работе алгоритма каждая сторона:

1. Генерирует случайное натуральное число – закрытый ключ.
2. Совместно с удаленной стороной устанавливают открытые параметры p и g(обычно значения p и g генерируются на одной стороне и передаются другой). Где число p:
   1. Является простым числом
   2. (p-1)/2 также должно быть простым числом.
   3. g является первообразным корнем по модулю p. (Также является простым числом).
3. Вычисляет открытый ключ, используя преобразования над закрытым ключом:
4. Обмениваются открытыми ключами с удаленной стороной
5. Вычисляют общий секретный ключ, используя открытый ключ другой стороны и свой закрытый ключ.

В практических реализациях для a и b используются числа порядка 10100 и p порядка 10300. Число g не обязано быть большим и обычно имеет значение в пределах первого десятка.

Криптографическая стойкость алгоритма Диффи-Хеллмана основана на предполагаемой сложности задачи дискретного логарифмирования. Все это сводится к тому, что задача Диффи-Хеллмана и задача дискретного логарифмирования считаются трудноразрешимыми.

Протокол Диффи — Хеллмана отлично противостоит пассивному нападению, но в случае реализации атаки «человек посередине» он не устоит. В самом деле, в протоколе ни Алиса, ни Боб не могут достоверно определить, кем является их собеседник, поэтому вполне возможно представить случай, при котором Боб и Алиса установили связь с Меллори, которая Алисе выдает себя за Боба, а Бобу представляется Алисой. То есть Меллори получает один общий ключ с Алисой (которая считает, что это Боб) и один общий ключ с Бобом (который считает, что это Алиса). А, следовательно, она может получать от Алисы любое сообщение для Боба, расшифровать его ключом, прочитать, зашифровать ключом и передать Бобу. Таким образом, подлог может оставаться незамеченным очень долгое время.

**2. Результаты проектирования программы.**

**Интерфейс**

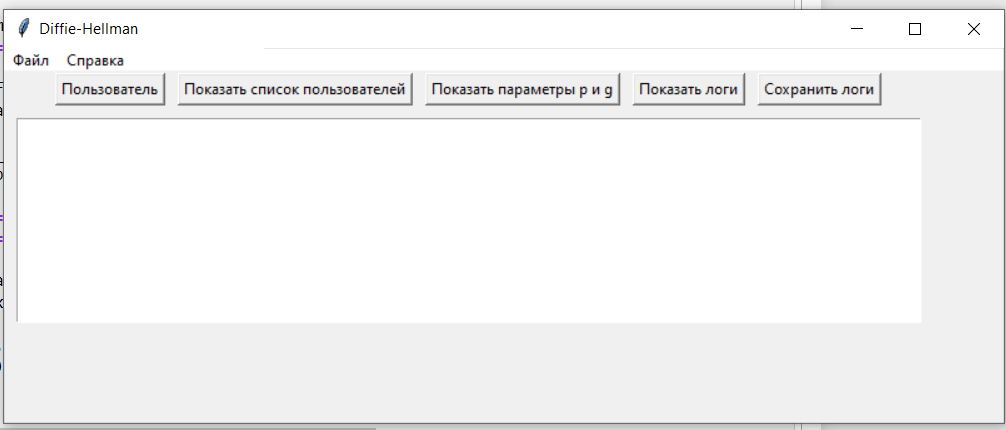


Рисунок 2.1. Главное окно программы.

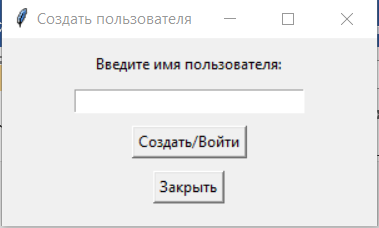


Рисунок 2.2. Окно создания нового пользователя.

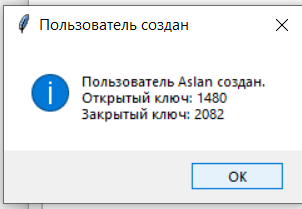


Рисунок 2.3. Окно успешного создания пользователя

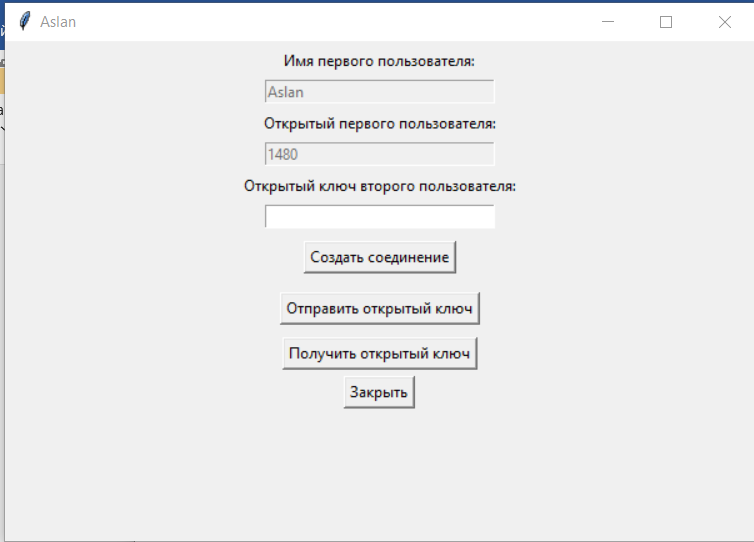


Рисунок 2.4. Окно создания одностороннего соединения

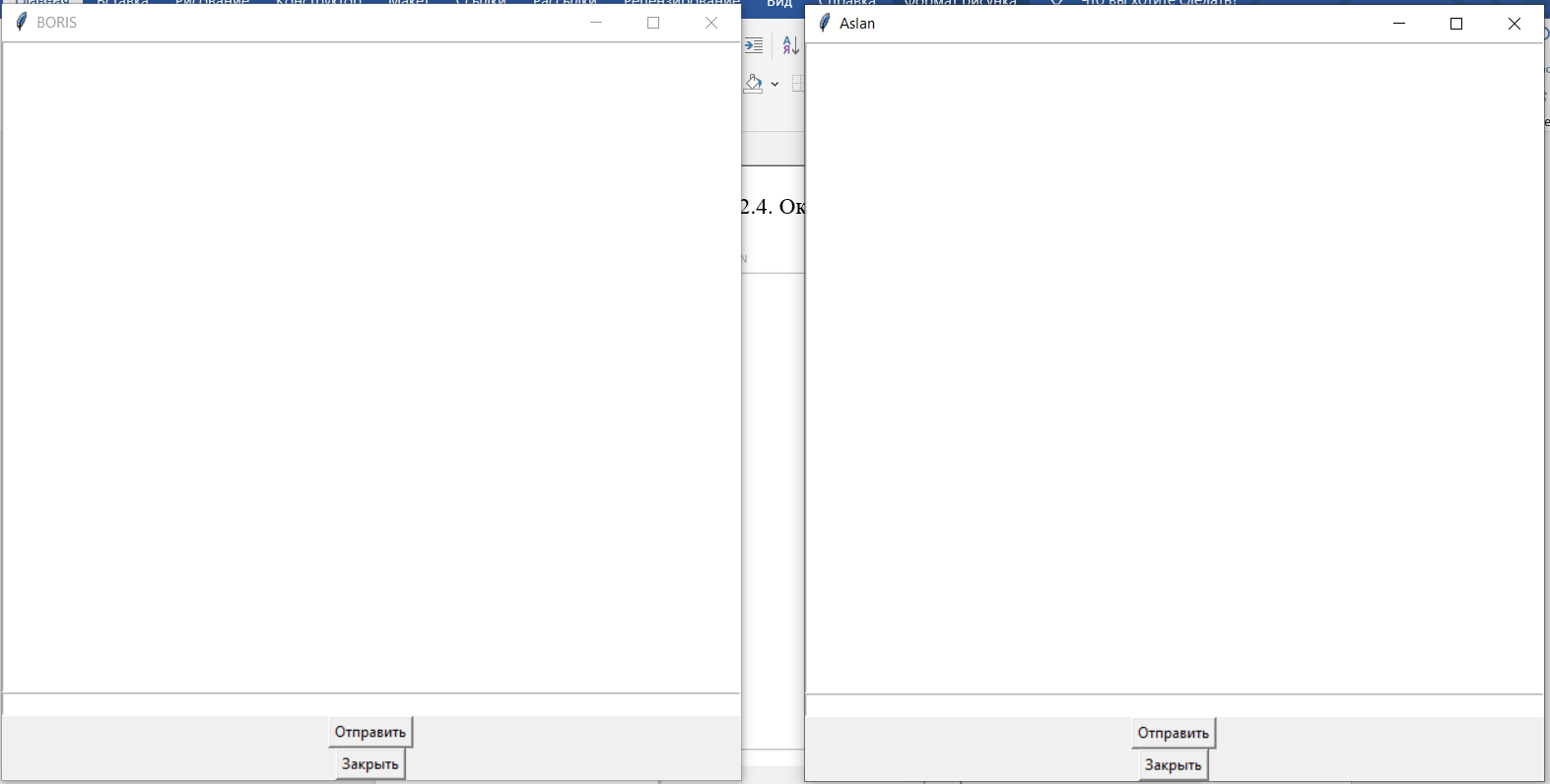


Рисунок 2.5. Окно чата пользователя

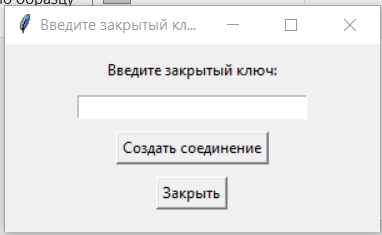


Рисунок 2.6. Окно входа пользователя

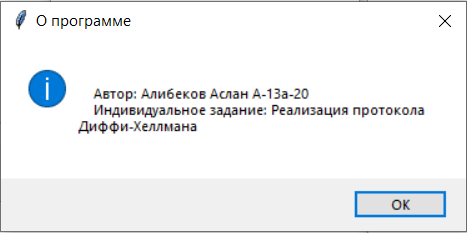


Рисунок 2.7. Окно о программе

При разработке интерфейса были предусмотрены сценарии в которых некоторые действия должны быть недоступны.

Нет возможности отправки сообщения, если у обоих пользователей не установлено соединение. После подключения обоих отправка сообщений становится доступной. При отключении одного из пользователей отправка сообщения снова становится недоступной.

Нельзя добавить пользователя, если такой уже создан. Закрытый ключ каждого пользователя уникален. При попытке входа под пользователем запрашивается закрытый ключ.

**3. Тестирование разработанной программы.**

Проверим работоспособность функции создания нового пользователя с именем Aslan.

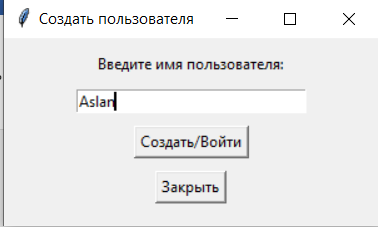


Рисунок 3.1. Создание пользователя Aslan.

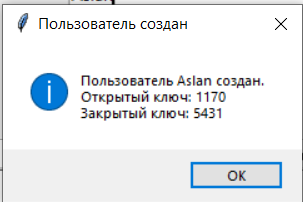


Рисунок 3.2. Начальное окно пользователя Aslan.

Окно пользователя с именем Aslan создано.

Открытый/Закрытый ключ пользователя Aslan сгенерирован и представлен пользователю.

Создадим второго пользователя

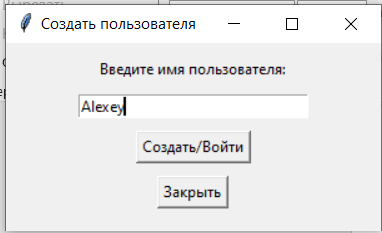


Рисунок 3.3. Создание пользователя Alexey.

Окно второго пользователя с именем Alexey успешно создано.

Открытый/Закрытый ключ пользователя Alexey сгенерирован и представлен пользователю.

Введём открытые ключи пользователей.

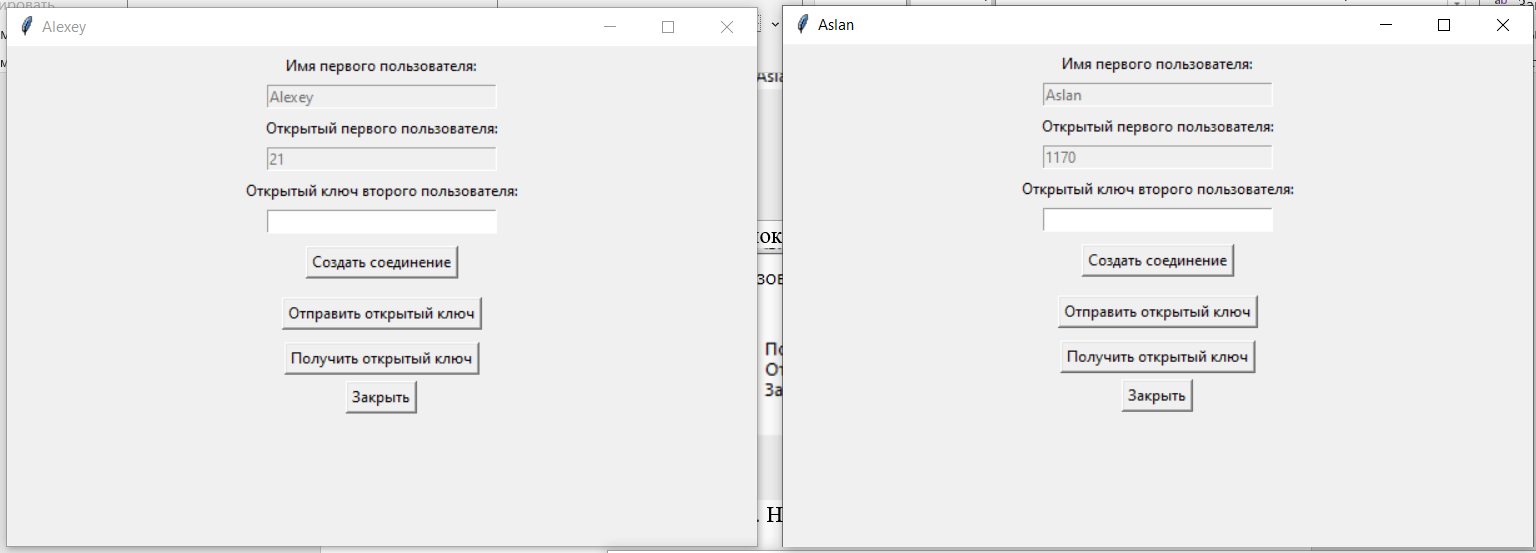


Рисунок 3.4. Окна пользователей для соединения.

Отправим и получим открытые ключи пользователей.

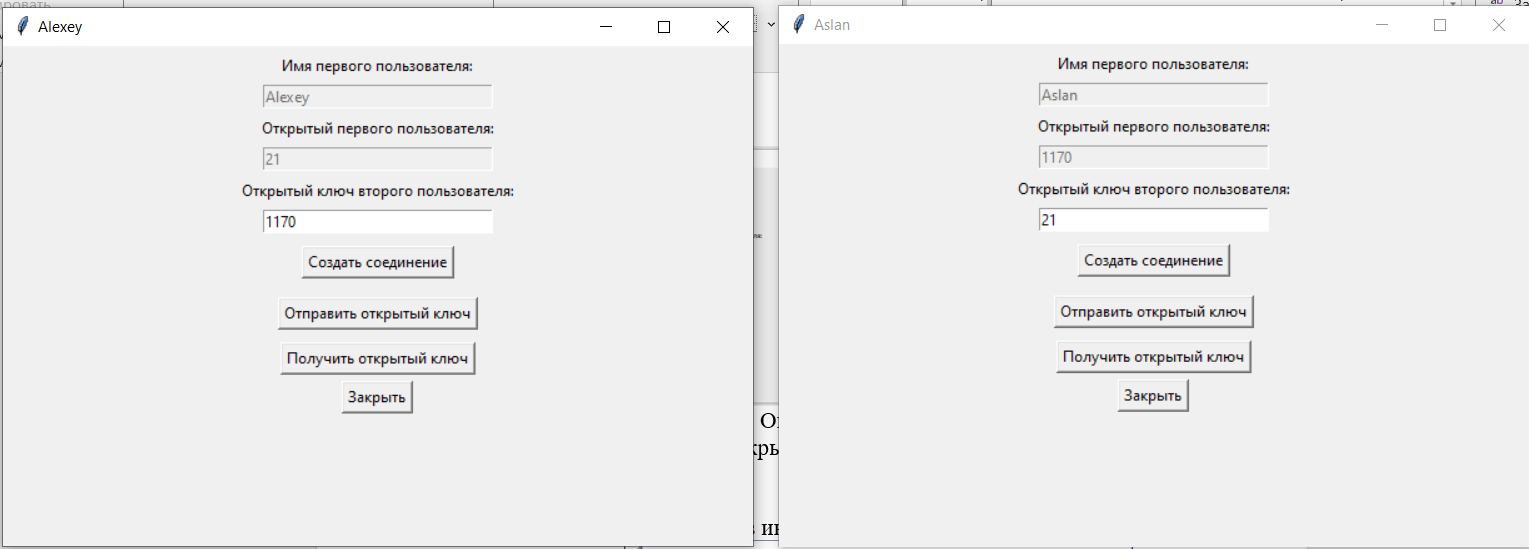


Рисунок 3.5. Окна пользователей с введенными открытыми ключами.

Установим соединение в интерфейсе пользователя Aslan.

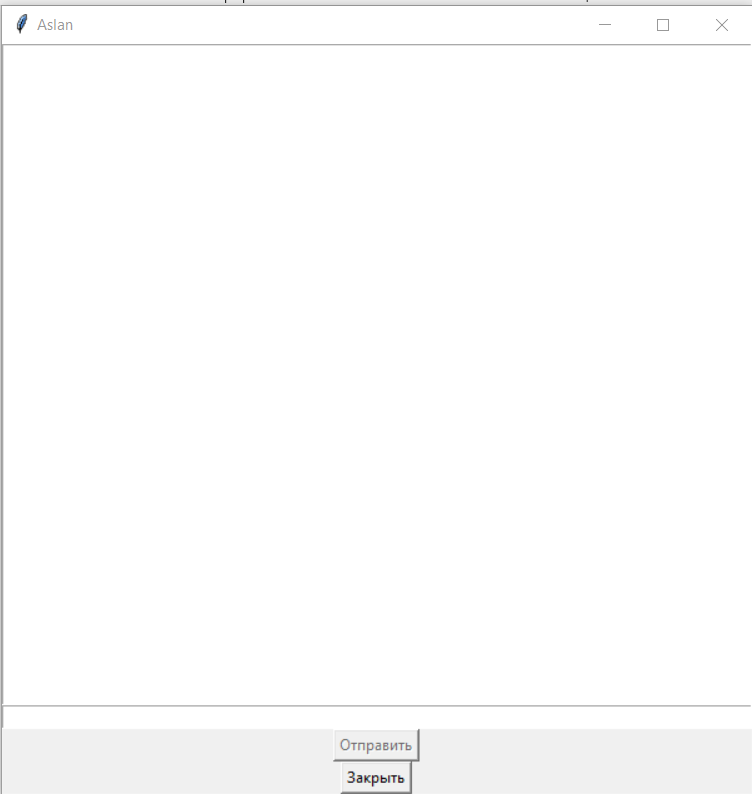


Рисунок 3.6. Окно пользователя Aslan после первоначальной установки соединения.

Соединение установлено, но отправка сообщений не доступна, потому что пользователь Alexey ещё не подключился.

Установим соединение в интерфейсе пользователя Alexey.

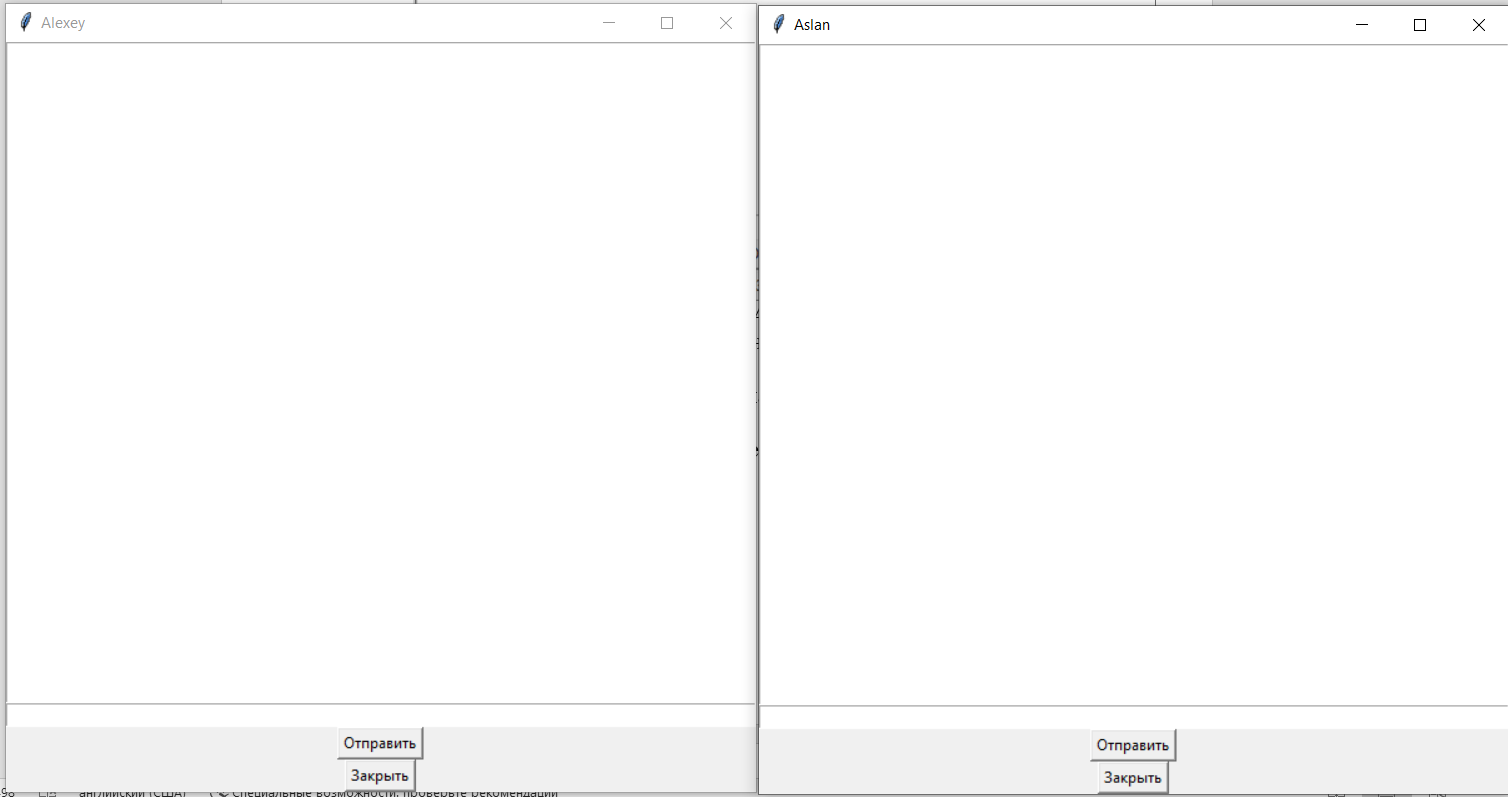


Рисунок 3.7. Чаты пользователей после первоначальной установки соединения.

Установка соединения произведена успешно.

Отправка сообщений доступна обоим пользователям.

Обменяемся сообщениями.

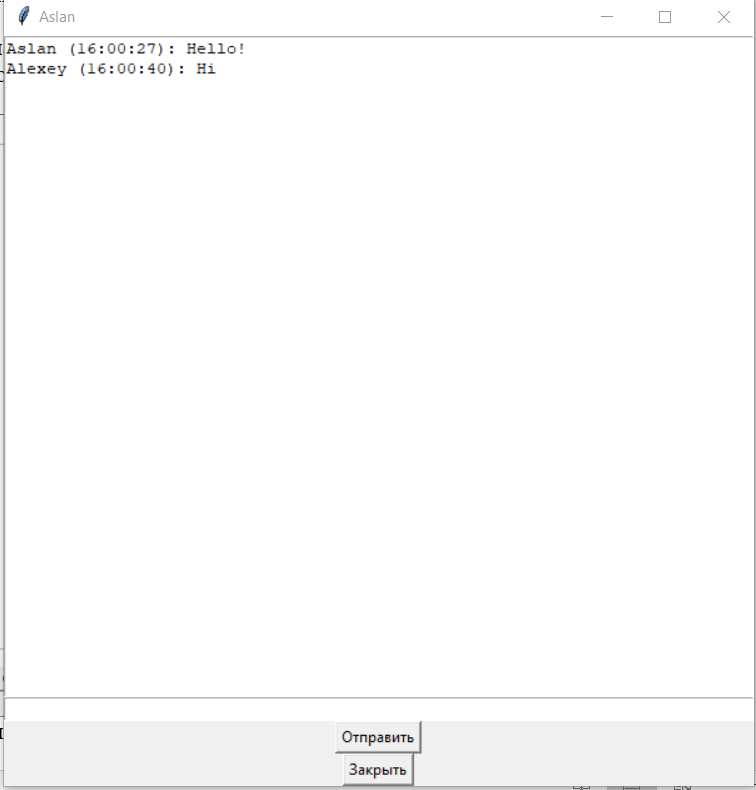


Рисунок 3.8. Окно пользователя Aslan после обмена сообщениями.

Отправка сообщений прошла успешно.

В интерфейсах пользователей корректно отображаются сообщения от собеседника и дублируются собственные сообщения.

Время отправки сообщений отображается корректно.

Произведём отключение соединения в интерфейсе пользователя Aslan.

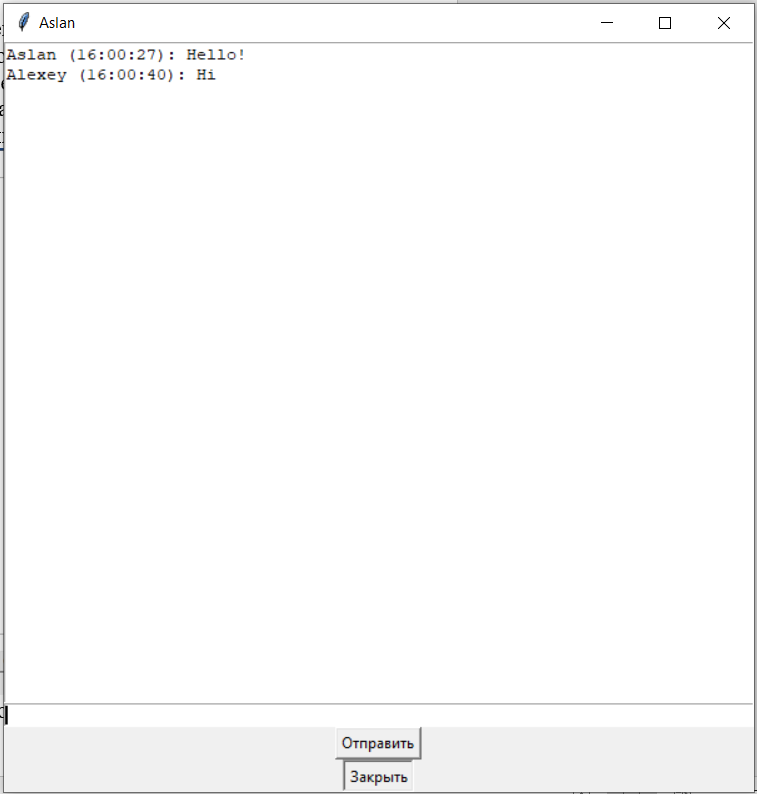


Рисунок 3.9. Пользователь Aslan отключился.

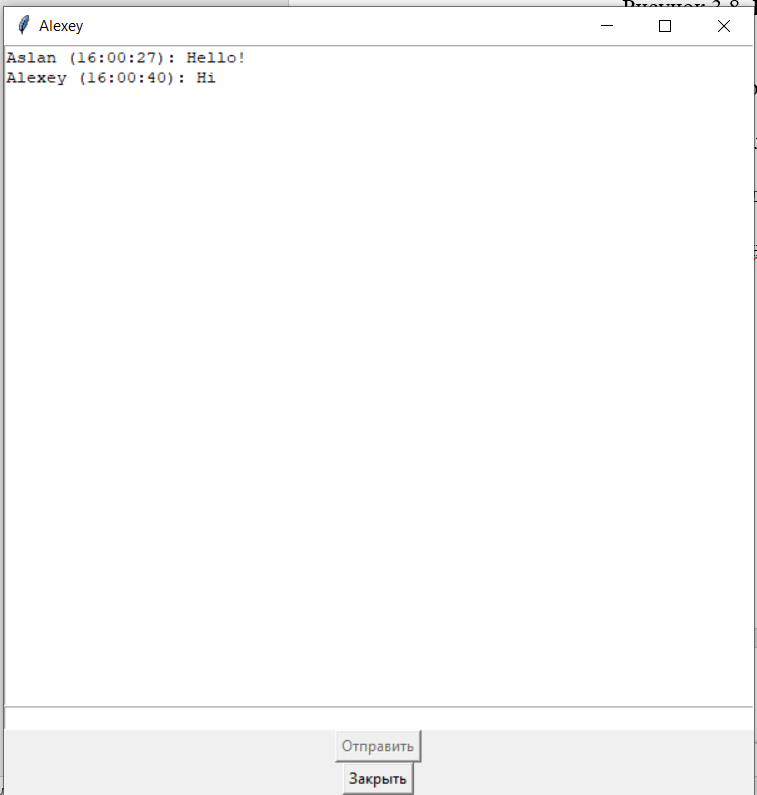


Рисунок 3.10. Окно пользователя Alexey после отключения пользователя Aslan.

При этом в интерфейсе пользователя Alexey кнопка отправка сообщений стала недоступна.

История диалога доступна для просмотра.

Произведём переподключение в интерфейсе пользователя Aslan.

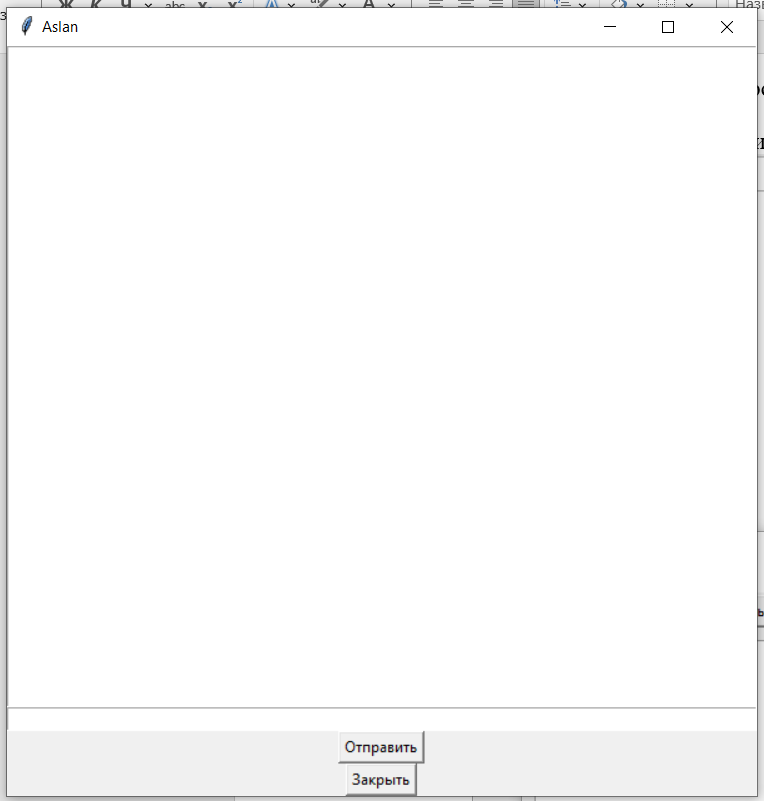


Рисунок 3.11. Окно пользователя Aslan после переподключения и отправки сообщения.

Подключение успешно установлено.

Отправим сообщение.

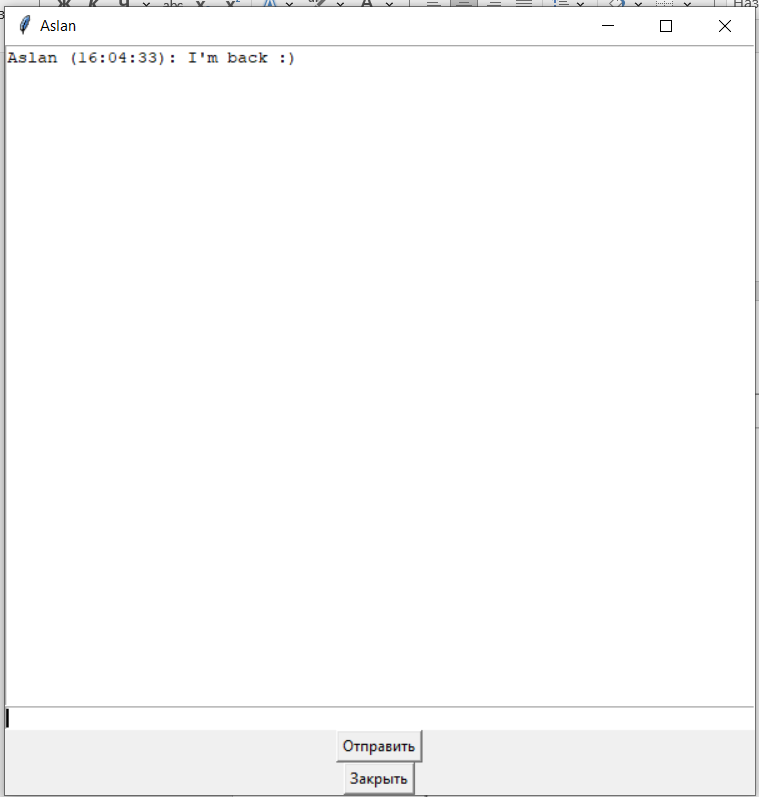


Рисунок 3.12. Aslan отправил сообщение.

Сообщение успешно отправлено.

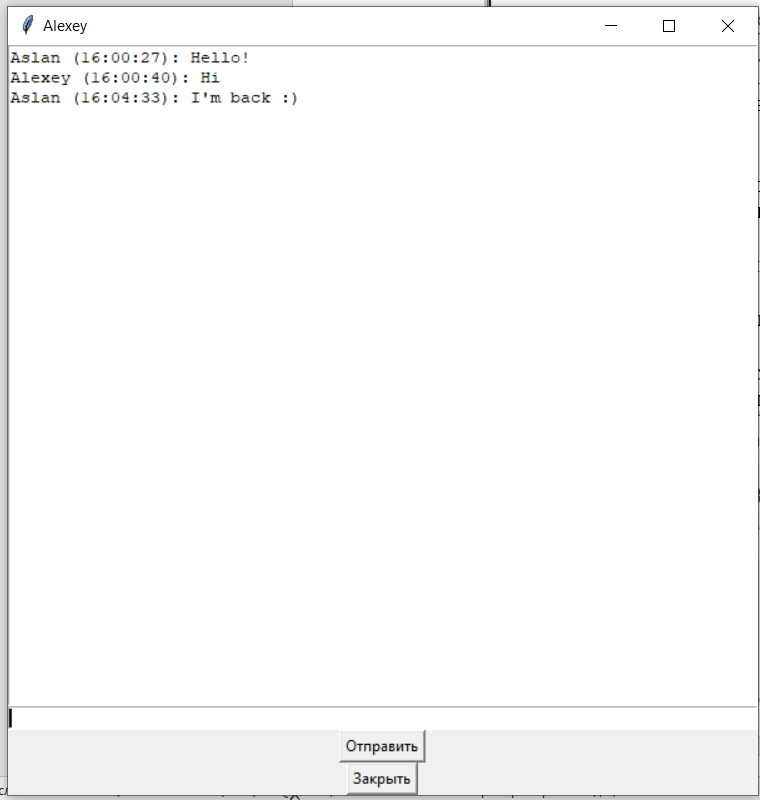


Рисунок 3.13. Окно пользователя Alexey после переподключения пользователя Aslan.

В интерфейсе пользователя Alexey стала доступной возможность отправки сообщений.

Новое сообщение пользователя Aslan получено корректно.

Произведём отключение соединения в интерфейсе пользователя Aslan и попробуем подключится с неверным закрытым ключом.

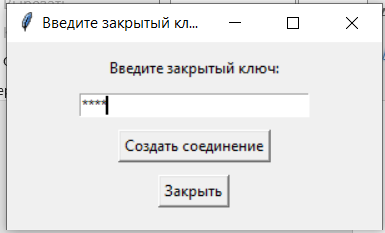


Рисунок 3.14. Окно ввода закрытого ключа пользователя Aslan.

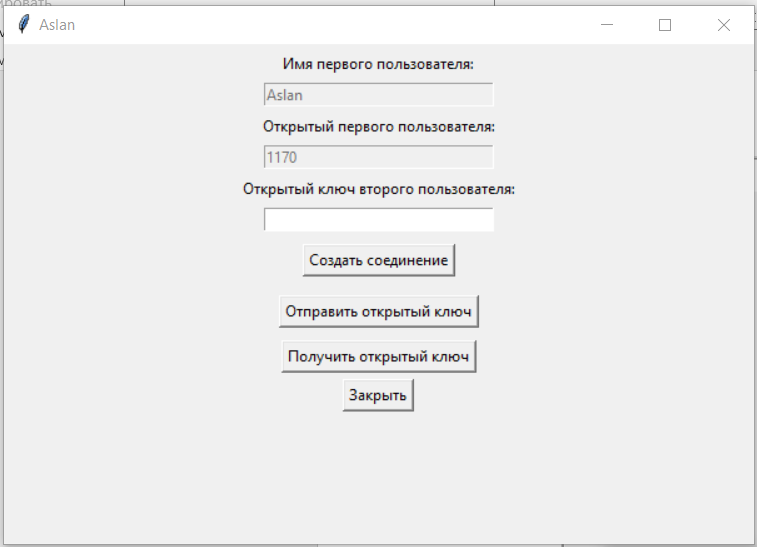


Рисунок 3.15. Окно создания соединения пользователя Aslan

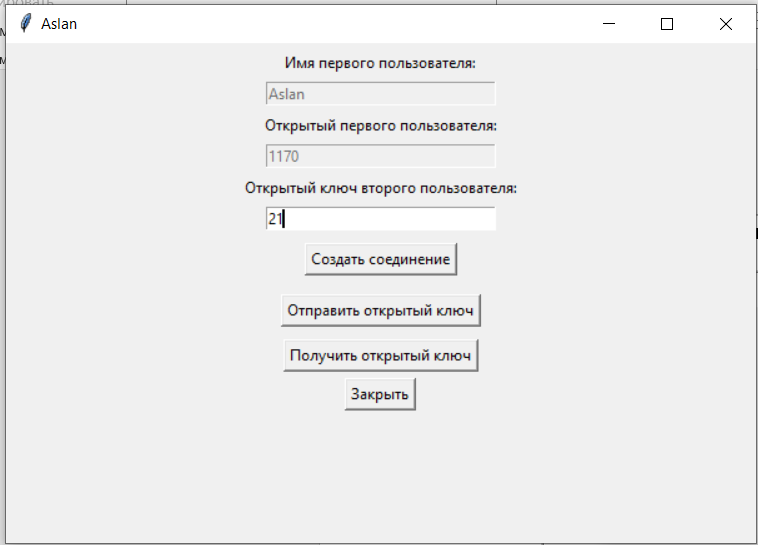


Рисунок 3.16. Окно ввода правильного открытого ключа Alexey после подключения по неверному закрытому ключу.

Подключение установлено.

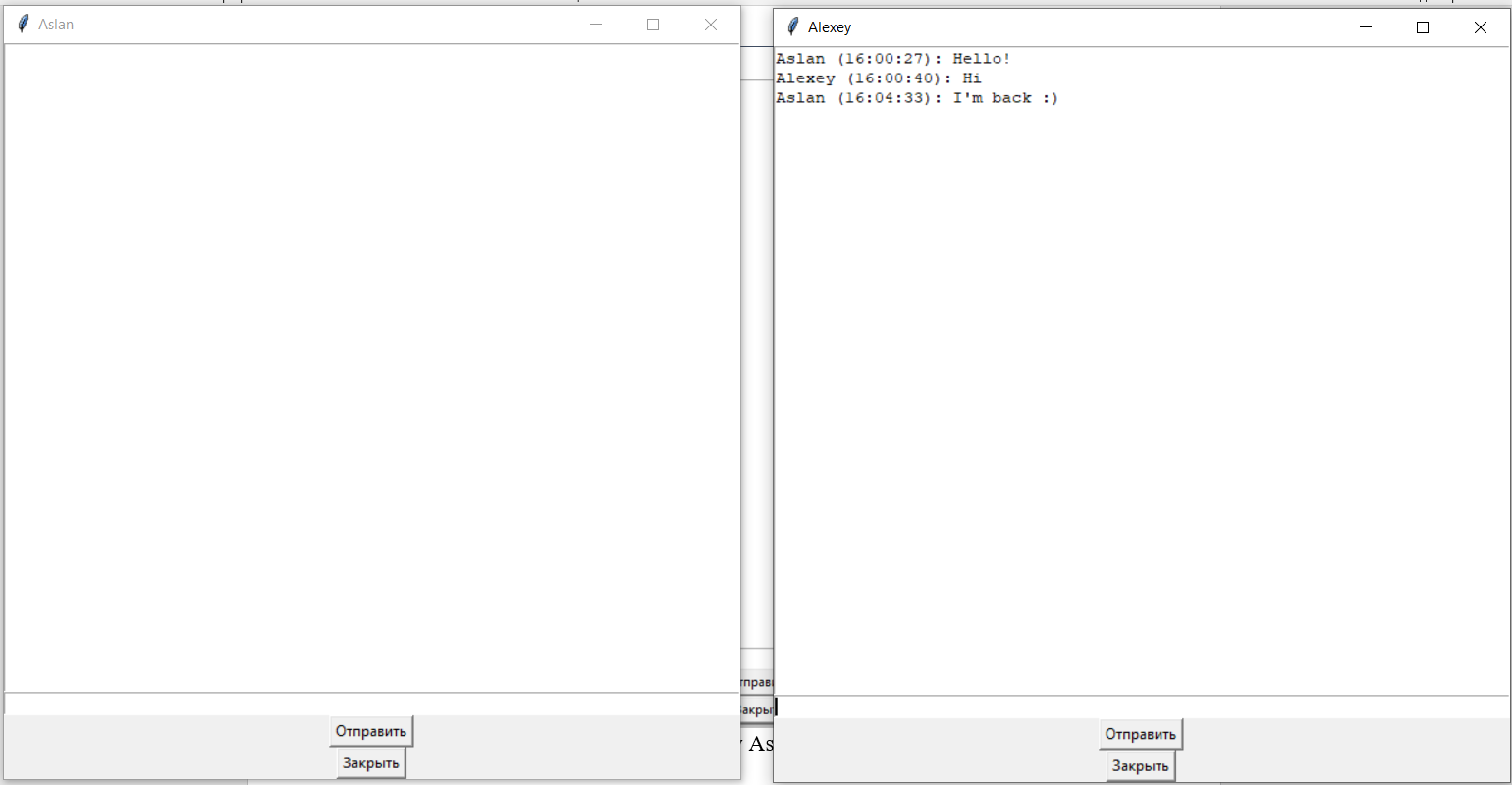


Рисунок 3.17. Окна чатов между Aslan и Alexey.

Отправим сообщение

Сообщение успешно отправлено.

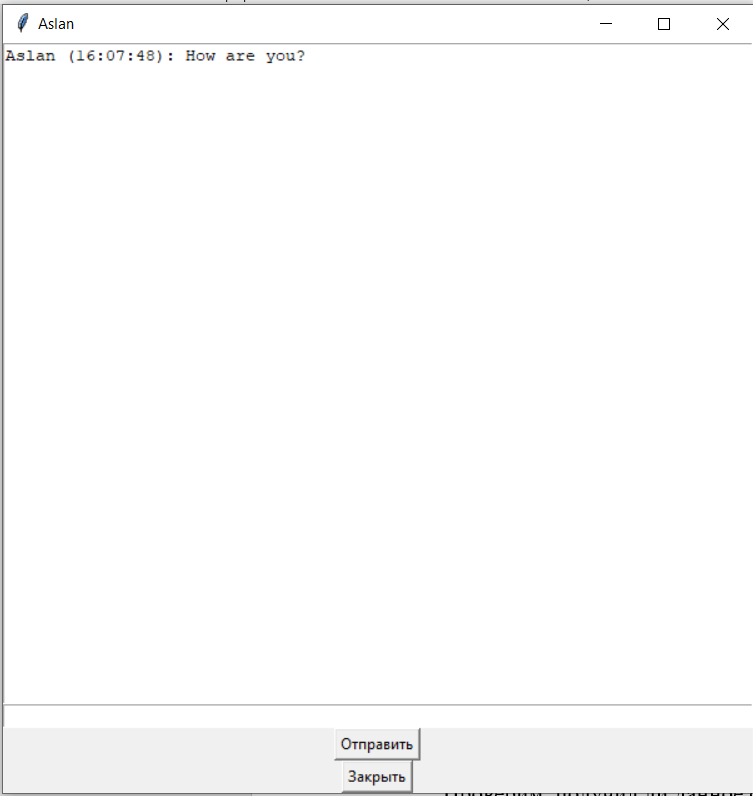


Рисунок 3.18. Отправленное сообщение от Aslan к Alexey.

Проверим, получил ли данное сообщение пользователь Alexey.

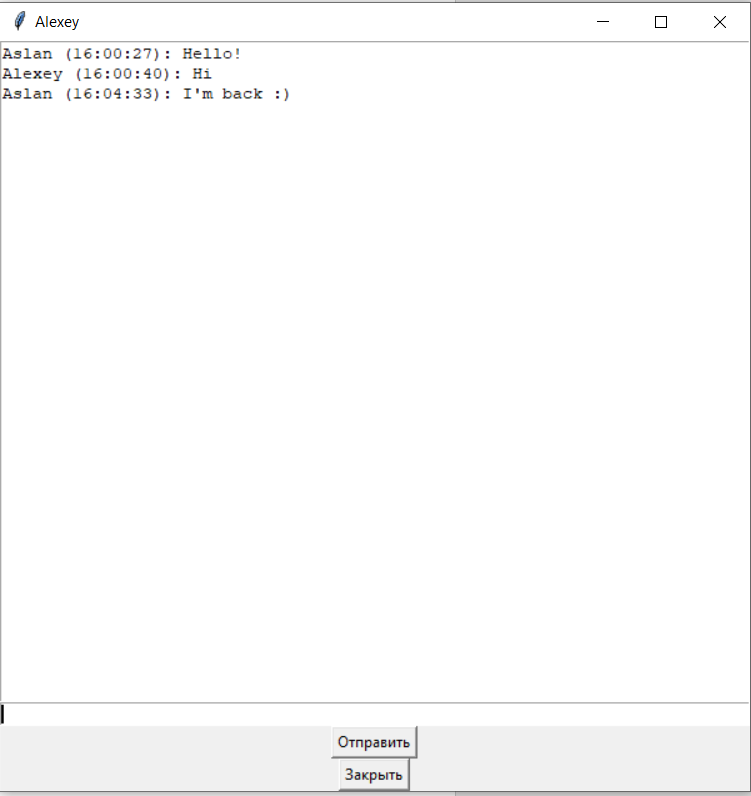
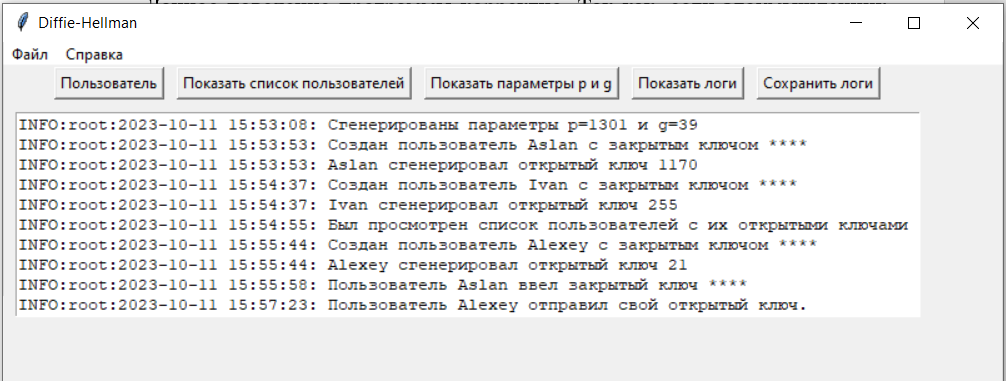


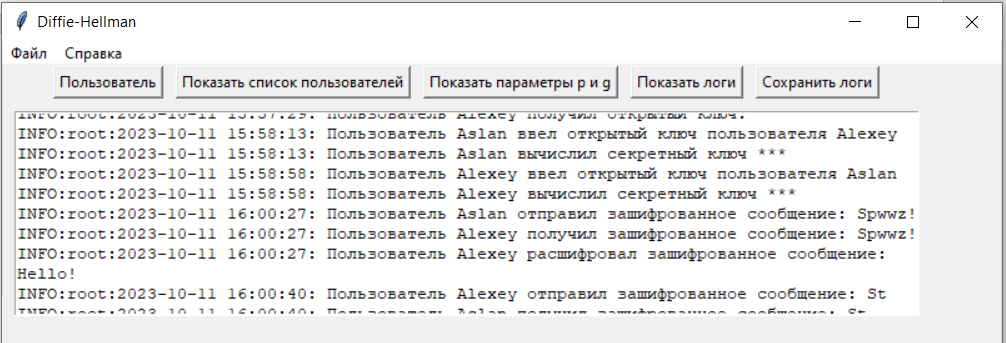
Рисунок 3.19. Окно пользователя Alexey после отправки пользователем Aslan сообщения при подключении по неверному ключу.

Пользователь Alexey не получил сообщение от пользователя Aslan, который ввёл неправильный закрытый ключ.

Данное поведение программы корректно. Так как, если злоумышленник произведёт подключение по неверному закрытому ключу, он не должен узнать, что ключ неверный, и программа должна имитировать работу конкретного сценария использования.

Проверим лог программы.





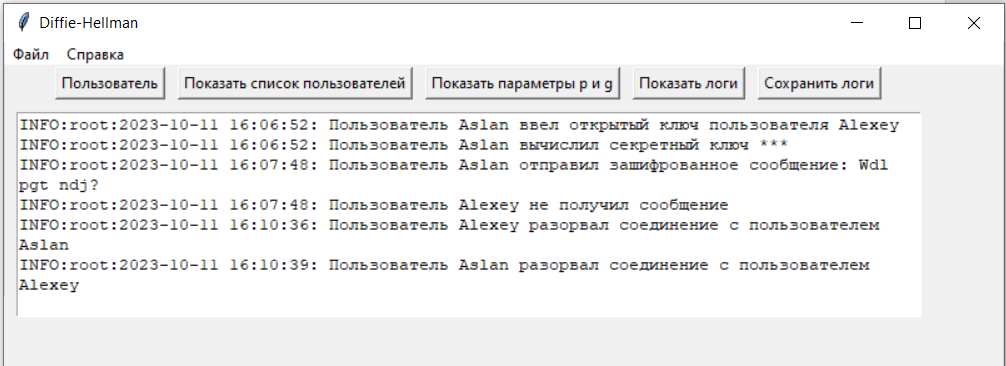


Рисунок 3.20. Логи действий программы.

Сохраним логи в файл.

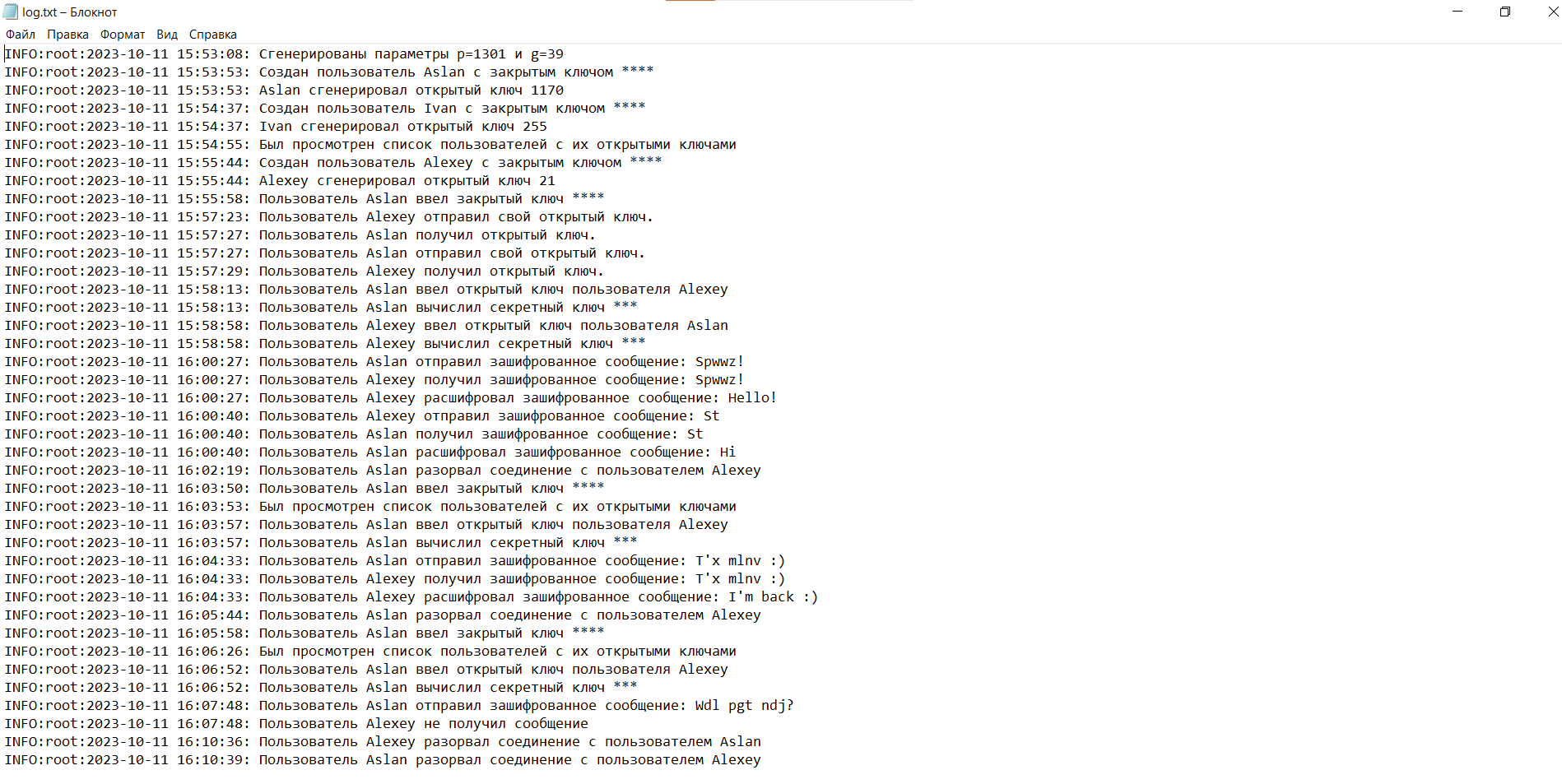


Рисунок 3.21. Сохранённые логи в файле.

Проверим окно Справки.

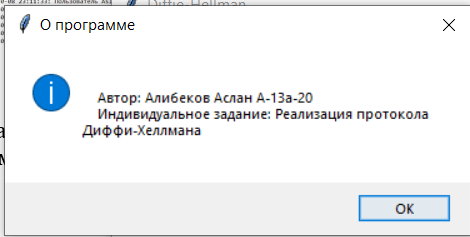


Рисунок 3.22. Окно справка.

Закроем программу.

Тестирование программы показало, что все компоненты и логика работают корректно.

**Заключение**

В результате выполнения работы, была создана программа, которая выполняет протокол Диффи-Хеллмана для генерации и обмена ключами с целью установки безопасного канала связи и обмена сообщениями. Также была предусмотрена функция логирования выполнения алгоритма в файле. Этот процесс позволил приобрести опыт в реализации криптографических алгоритмов и разработке программного обеспечения.

**Список литературы**

1. Уязвимость и защита алгоритма Диффи-Хеллмана

<https://elar.urfu.ru/bitstream/10995/46472/1/bip_2014_49.pdf>

1. Протокол Диффи — Хеллмана.

[https://ru.wikipedia.org/wiki/Протокол\_Диффи\_—\_Хеллмана](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB_%D0%94%D0%B8%D1%84%D1%84%D0%B8_%E2%80%94_%D0%A5%D0%B5%D0%BB%D0%BB%D0%BC%D0%B0%D0%BD%D0%B0)

1. Протокол Диффи — Хеллмана.

<https://habr.com/ru/articles/726324/>

**Приложение**

**import** tkinter **as** tk

**import** tkinter**.**messagebox **as** messagebox

**from** tkinter **import** simpledialog

**import** random

**import** json

**from** datetime **import** datetime

**from** sympy **import** primerange

**import** logging

**import** string

**from** tkinter **import** Text

**from** math **import** gcd **as** bltin\_gcd

**import** ctypes

logging**.**basicConfig**(**level**=**logging**.**INFO**)**

**def** log\_action**(**action**):**

current\_time **=** datetime**.**now**().**strftime**(**"%Y-%m-%d %H:%M:%S"**)**

log\_entry **=** f"{current\_time}: {action}\n"

log\_text**.**insert**(**tk**.**END**,** log\_entry**)**

**def** display\_logs**():**

log\_filename **=** simpledialog**.**askstring**(**"Показать логи"**,** "Введите название текстового файла для просмотра логов:"**)**

**if** log\_filename**:**

**try:**

**with** **open(**log\_filename**,** 'r'**)** **as** log\_file**:**

log\_content **=** log\_file**.**read**()**

log\_display **=** tk**.**Toplevel**(**root**)**

log\_display**.**title**(**"Логи"**)**

log\_display**.**geometry**(**"800x400"**)**

log\_text **=** tk**.**Text**(**log\_display**,** wrap**=**tk**.**WORD**,** width**=**80**,** height**=**20**)**

log\_text**.**insert**(**tk**.**END**,** log\_content**)**

log\_text**.**pack**()**

**except** **Exception** **as** e**:**

messagebox**.**showerror**(**"Ошибка"**,** f"Ошибка при открытии файла с логами: {**str(**e**)**}"**)**

**def** save\_logs\_to\_file**():**

logs **=** log\_text**.**get**(**1.0**,** tk**.**END**)**

filename **=** simpledialog**.**askstring**(**"Сохранить логи"**,** "Введите название текстового файла для сохранения логов:"**)**

**if** filename**:**

filename **=** filename **+** ".txt"

**try:**

**with** **open(**filename**,** 'w'**)** **as** log\_file**:**

log\_file**.**write**(**logs**)**

log\_action**(**f"Логи были сохранены в файл: {filename}"**)**

messagebox**.**showinfo**(**"Сохранение логов"**,** "Логи были успешно сохранены."**)**

**except** **Exception** **as** e**:**

messagebox**.**showerror**(**"Ошибка"**,** f"Ошибка при сохранении логов: {**str(**e**)**}"**)**

**def** show\_about**():**

about\_info **=** """

Автор: Алибеков Аслан А-13а-20

Индивидуальное задание: Реализация протокола Диффи-Хеллмана

"""

tk**.**messagebox**.**showinfo**(**"О программе"**,** about\_info**)**

**def** primRoots**(**modulo**):**

required\_set **=** **{**num **for** num **in** **range(**1**,** modulo**)** **if** bltin\_gcd**(**num**,** modulo**)}**

primitive\_roots **=** **[]**

**for** g **in** **range(**1**,** modulo**):**

**if** required\_set **==** **{pow(**g**,** powers**,** modulo**)** **for** powers **in** **range(**1**,** modulo**)}:**

primitive\_roots**.**append**(**g**)**

**if** **len(**primitive\_roots**)** **==** 50**:**

**break**

**return** primitive\_roots

**def** generate\_DH\_parameters**():**

parameters **=** load\_DH\_parameters\_from\_file**(**'DH\_parameters.json'**)**

**if** parameters **is** **not** **None:**

p**,** g **=** parameters

**else:**

primes **=** **list(**primerange**(**1000**,** 5000**))**

p **=** random**.**choice**(**primes**)**

g **=** random**.**choice**(**primRoots**(**p**))**

save\_DH\_parameters\_to\_file**(**'DH\_parameters.json'**,** p**,** g**)**

log\_action**(**f"Сгенерированы параметры p={p} и g={g}"**)**

**return** p**,** g

**def** save\_DH\_parameters\_to\_file**(**filename**,** p**,** g**):**

parameters **=** **{**'p'**:** p**,** 'g'**:** g**}**

**with** **open(**filename**,** 'w'**)** **as** file**:**

json**.**dump**(**parameters**,** file**)**

**def** load\_DH\_parameters\_from\_file**(**filename**):**

**try:**

**with** **open(**filename**,** 'r'**)** **as** file**:**

parameters **=** json**.**load**(**file**)**

p **=** parameters**[**'p'**]**

g **=** parameters**[**'g'**]**

**return** p**,** g

**except** **FileNotFoundError:**

**return** **None**

**def** show\_DH\_parameters**():**

parameters **=** load\_DH\_parameters\_from\_file**(**'DH\_parameters.json'**)**

**if** parameters **is** **not** **None:**

p**,** g **=** parameters

messagebox**.**showinfo**(**"Параметры DH"**,** f"p: {p}\ng: {g}"**)**

log\_action**(**"Были просмотрены параметры p и g"**)**

**else:**

messagebox**.**showerror**(**"Ошибка"**,** "Параметры DH не найдены."**)**

**def** generate\_keys**(**users**):**

key **=** random**.**randint**(**10000**,** 300000**)**

**while** key **in** users**.**values**():**

key **=** random**.**randint**(**10000**,** 300000**)**

**return** key

**def** save\_users\_to\_file**(**filename**,** users**):**

**with** **open(**filename**,** 'w'**)** **as** file**:**

json**.**dump**(**users**,** file**)**

**def** load\_users\_from\_file**(**filename**):**

**try:**

**with** **open(**filename**,** 'r'**)** **as** file**:**

**return** json**.**load**(**file**)**

**except** **FileNotFoundError:**

**return** **{}**

**def** create\_private\_key\_window**(**users**,** connections**,** chat\_messages**,** chat\_displays**,** existing\_user\_name**,** user\_secret\_keys**,**

connection\_windows**):**

private\_key\_window **=** tk**.**Toplevel**(**root**)**

private\_key\_window**.**title**(**"Введите закрытый ключ"**)**

private\_key\_window**.**geometry**(**"300x150"**)**

**def** create\_connection\_with\_private\_key**(**users**,** connections**,** chat\_messages**,** chat\_displays**,** existing\_user\_name**,**

user\_secret\_keys**,** connection\_windows**):**

user\_private\_key **=** private\_key\_entry**.**get**().**strip**()**

log\_action**(**f"Пользователь {existing\_user\_name} ввел закрытый ключ {'\*' **\*** **len(str(**user\_private\_key**))**}"**)**

**if** user\_private\_key**.**isdigit**():**

user\_private\_key **=** **int(**user\_private\_key**)**

**if** user\_private\_key**:**

**if** existing\_user\_name **in** users**:**

user\_public\_key **=** users**[**existing\_user\_name**]**

create\_connection\_window**(**users**,** connections**,** chat\_messages**,** chat\_displays**,** existing\_user\_name**,**

user\_public\_key**,** user\_private\_key**,** user\_secret\_keys**,** connection\_windows**)**

private\_key\_window**.**destroy**()**

**else:**

messagebox**.**showerror**(**"Ошибка"**,** "Пользователь не существует."**)**

**else:**

messagebox**.**showerror**(**"Ошибка"**,** "Закрытый ключ не введен."**)**

**else:**

messagebox**.**showerror**(**"Ошибка"**,** "Введите корректный закрытый ключ."**)**

private\_key\_label **=** tk**.**Label**(**private\_key\_window**,** text**=**"Закрытый ключ:"**)**

private\_key\_entry **=** tk**.**Entry**(**private\_key\_window**,** width**=**30**,** show**=**'\*'**)**

create\_connection\_button **=** tk**.**Button**(**private\_key\_window**,** text**=**"Создать соединение"**,**

command**=lambda:** create\_connection\_with\_private\_key**(**users**,** connections**,**

chat\_messages**,**

chat\_displays**,**

existing\_user\_name**,**

user\_secret\_keys**,**

connection\_windows**))**

close\_private\_key\_window\_button **=** tk**.**Button**(**private\_key\_window**,** text**=**"Закрыть"**,** command**=**private\_key\_window**.**destroy**)**

private\_key\_label**.**pack**(**pady**=**10**)**

private\_key\_entry**.**pack**()**

create\_connection\_button**.**pack**(**pady**=**10**)**

close\_private\_key\_window\_button**.**pack**()**

**def** create\_user\_window**(**users**,** connections**,** chat\_messages**,** chat\_displays**,** user\_secret\_keys**,** connection\_windows**):**

user\_window **=** tk**.**Toplevel**(**root**)**

user\_window**.**title**(**"Создать пользователя"**)**

user\_window**.**geometry**(**"300x150"**)**

**def** create\_user\_and\_connection**(**users**,** connections**,** chat\_messages**,** chat\_displays**,** user\_secret\_keys**,**

connection\_windows**):**

user\_name **=** user\_name\_entry**.**get**().**strip**()**

**if** user\_name**:**

**if** user\_name **not** **in** users**.**keys**():**

**if** **len(**users**)** **==** 0**:**

p**,** g **=** generate\_DH\_parameters**()**

save\_DH\_parameters\_to\_file**(**'DH\_parameters.json'**,** p**,** g**)**

**else:**

p**,** g **=** load\_DH\_parameters\_from\_file**(**'DH\_parameters.json'**)**

user\_private\_key **=** generate\_keys**(**users**)**

log\_action**(**f"Создан пользователь {user\_name} с закрытым ключом {'\*' **\*** **len(str(**user\_private\_key**))**}"**)**

user\_public\_key **=** **(**g **\*\*** user\_private\_key**)** **%** p

users**[**user\_name**]** **=** user\_public\_key

save\_users\_to\_file**(**users\_filename**,** users**)**

log\_action**(**f"{user\_name} сгенерировал открытый ключ {user\_public\_key}"**)**

messagebox**.**showinfo**(**"Пользователь создан"**,**

f"Пользователь {user\_name} создан.\nОткрытый ключ: {user\_public\_key}\nЗакрытый ключ: {user\_private\_key}"**)**

create\_connection\_window**(**users**,** connections**,** chat\_messages**,** chat\_displays**,** user\_name**,** user\_public\_key**,**

user\_private\_key**,** user\_secret\_keys**,** connection\_windows**)**

user\_window**.**destroy**()**

**else:**

user\_window**.**destroy**()**

create\_private\_key\_window**(**users**,** connections**,** chat\_messages**,** chat\_displays**,** user\_name**,** user\_secret\_keys**,**

connection\_windows**)**

**else:**

messagebox**.**showerror**(**"Ошибка"**,** "Введите имя пользователя."**)**

user\_name\_label **=** tk**.**Label**(**user\_window**,** text**=**"Имя пользователя:"**)**

user\_name\_entry **=** tk**.**Entry**(**user\_window**,** width**=**30**)**

create\_button **=** tk**.**Button**(**user\_window**,** text**=**"Создать/Войти"**,**

command**=lambda:** create\_user\_and\_connection**(**users**,** connections**,** chat\_messages**,**

chat\_displays**,** user\_secret\_keys**,**

connection\_windows**))**

close\_button **=** tk**.**Button**(**user\_window**,** text**=**"Закрыть"**,** command**=**user\_window**.**destroy**)**

user\_name\_label**.**pack**(**pady**=**10**)**

user\_name\_entry**.**pack**()**

create\_button**.**pack**(**pady**=**10**)**

close\_button**.**pack**()**

**def** show\_user\_list**(**users**):**

user\_list\_window **=** tk**.**Toplevel**(**root**)**

user\_list\_window**.**title**(**"Список пользователей"**)**

user\_list\_window**.**geometry**(**"400x300"**)**

user\_list\_label **=** tk**.**Label**(**user\_list\_window**,** text**=**"Список пользователей:"**)**

user\_list\_label**.**pack**()**

users\_data **=** load\_users\_from\_file**(**users\_filename**)**

log\_action**(**"Был просмотрен список пользователей с их открытыми ключами"**)**

**for** user**,** data **in** users\_data**.**items**():**

user\_info\_label **=** tk**.**Label**(**user\_list\_window**,** text**=**f"Пользователь: {user}, Открытый ключ: {data}"**)**

user\_info\_label**.**pack**()**

**def** create\_connection\_window**(**users**,** connections**,** chat\_messages**,** chat\_displays**,** user1\_name**,** user1\_key**,** user1\_private\_key**,**

user\_secret\_keys**,** connection\_windows**):**

connection\_window **=** tk**.**Toplevel**(**root**)**

connection\_window**.**title**(**f"{user1\_name}"**)**

connection\_window**.**geometry**(**"600x400"**)**

connection\_windows**[**user1\_name**]** **=** connection\_window

**def** create\_connection**(**users**,** connections**,** chat\_messages**,** chat\_displays**,** user1\_name**,** user1\_key**,** user1\_private\_key**,**

user\_secret\_keys**):**

user2\_key **=** user2\_key\_entry**.**get**()**

p**,** g **=** load\_DH\_parameters\_from\_file**(**'DH\_parameters.json'**)**

**if** user1\_name **in** users **and** **int(**user1\_key**)** **==** users**[**user1\_name**]:**

**if** user2\_key**.**isdigit**()** **and** **int(**user2\_key**)** **in** users**.**values**():**

user2\_name **=** **[**name **for** name**,** key **in** users**.**items**()** **if** key **==** **int(**user2\_key**)][**0**]**

key\_secret **=** **int(**user2\_key**)** **\*\*** user1\_private\_key **%** p

log\_action**(**f"Пользователь {user1\_name} ввел открытый ключ пользователя {user2\_name}"**)**

log\_action**(**f"Пользователь {user1\_name} вычислил секретный ключ {'\*' **\*** **len(str(**key\_secret**))**}"**)**

user\_secret\_keys**[**user1\_name**]** **=** key\_secret

create\_chat**(**user1\_name**,** user2\_name**,** chat\_messages**,** chat\_displays**,** user\_secret\_keys**)**

**if** user1\_name **in** connections**:**

connections**[**user1\_name**].**append**(**user2\_name**)**

**else:**

connections**[**user1\_name**]** **=** **[**user2\_name**]**

connection\_window**.**destroy**()**

**else:**

messagebox**.**showerror**(**"Ошибка"**,** "Неверный ключ второго пользователя."**)**

**else:**

messagebox**.**showerror**(**"Ошибка"**,** "Неверное имя пользователя или ключ."**)**

key **=** **None**

**def** send\_public\_key**(**user1\_name**,** user1\_key**,** connection\_windows**):**

**global** key

first\_user\_window **=** connection\_windows**.**get**(**user1\_name**)**

**if** first\_user\_window**:**

key **=** user1\_key

log\_action**(**f"Пользователь {user1\_name} отправил свой открытый ключ."**)**

**return** key

**def** receive\_public\_key**(**user1\_key**,** user1\_name**,** connection\_windows**):**

**global** key

**try:**

**if** key**:**

second\_user\_window **=** connection\_windows**.**get**(**user1\_name**)**

**if** second\_user\_window **and** key **!=** user1\_key**:**

log\_action**(**f"Пользователь {user1\_name} получил открытый ключ."**)**

user2\_key\_entry**.**delete**(**0**,** tk**.**END**)**

user2\_key\_entry**.**insert**(**0**,** **str(**key**))**

key **=** **None**

**except** **NameError:**

**pass**

user1\_name\_label **=** tk**.**Label**(**connection\_window**,** text**=**"Имя первого пользователя:"**)**

user1\_name\_entry **=** tk**.**Entry**(**connection\_window**,** width**=**30**)**

user1\_name\_entry**.**insert**(**0**,** user1\_name**)**

user1\_name\_entry**.**config**(**state**=**tk**.**DISABLED**)**

user1\_key\_label **=** tk**.**Label**(**connection\_window**,** text**=**"Открытый первого пользователя:"**)**

user1\_key\_entry **=** tk**.**Entry**(**connection\_window**,** width**=**30**)**

user1\_key\_entry**.**insert**(**0**,** user1\_key**)**

user1\_key\_entry**.**config**(**state**=**tk**.**DISABLED**)**

user2\_key\_label **=** tk**.**Label**(**connection\_window**,** text**=**"Открытый ключ второго пользователя:"**)**

user2\_key\_entry **=** tk**.**Entry**(**connection\_window**,** width**=**30**)**

create\_connection\_button **=** tk**.**Button**(**connection\_window**,** text**=**"Создать соединение"**,**

command**=lambda:** create\_connection**(**users**,** connections**,** chat\_messages**,**

chat\_displays**,** user1\_name**,** user1\_key**,**

user1\_private\_key**,** user\_secret\_keys**))**

send\_public\_key\_button **=** tk**.**Button**(**connection\_window**,** text**=**"Отправить открытый ключ"**,**

command**=lambda:** send\_public\_key**(**user1\_name**,** user1\_key**,** connection\_windows**))**

receive\_public\_key\_button **=** tk**.**Button**(**connection\_window**,** text**=**"Получить открытый ключ"**,**

command**=lambda:** receive\_public\_key**(**user1\_key**,** user1\_name**,** connection\_windows**))**

**def** check\_user\_count\_for\_button**(**users**,** create\_connection\_button**):**

**if** connection\_window**.**winfo\_exists**():**

**try:**

**if** **len(**users**)** **>=** 2**:**

create\_connection\_button**.**config**(**state**=**tk**.**NORMAL**)**

**else:**

create\_connection\_button**.**config**(**state**=**tk**.**DISABLED**)**

**except** **KeyError:**

create\_connection\_button**.**config**(**state**=**tk**.**DISABLED**)**

connection\_window**.**after**(**1000**,** **lambda:** check\_user\_count\_for\_button**(**users**,** create\_connection\_button**))**

check\_user\_count\_for\_button**(**users**,** create\_connection\_button**)**

close\_connection\_button **=** tk**.**Button**(**connection\_window**,** text**=**"Закрыть"**,** command**=**connection\_window**.**destroy**)**

user1\_name\_label**.**pack**(**pady**=**5**)**

user1\_name\_entry**.**pack**()**

user1\_key\_label**.**pack**(**pady**=**5**)**

user1\_key\_entry**.**pack**()**

user2\_key\_label**.**pack**(**pady**=**5**)**

user2\_key\_entry**.**pack**()**

create\_connection\_button**.**pack**(**pady**=**10**)**

send\_public\_key\_button**.**pack**(**pady**=**5**)**

receive\_public\_key\_button**.**pack**(**pady**=**5**)**

close\_connection\_button**.**pack**()**

**def** are\_connected**(**user1\_name**,** user2\_name**,** connections**):**

**if** user1\_name **in** connections**[**user2\_name**]** **and** user2\_name **in** connections**[**user1\_name**]:**

**return** **True**

**return** **False**

**def** caesar\_cipher**(**message**,** shift**):**

result **=** ""

rus\_alphabet **=** 'абвгдеёжзийклмнопрстуфхцчшщъыьэюя'

lat\_alphabet **=** 'abcdefghijklmnopqrstuvwxyz'

digits **=** '0123456789'

**for** char **in** message**:**

**if** char**.**isalpha**():**

alphabet **=** lat\_alphabet **if** char**.**lower**()** **in** lat\_alphabet **else** rus\_alphabet

shift\_amount **=** shift **%** **len(**alphabet**)**

is\_upper **=** char**.**isupper**()**

shifted\_char **=** alphabet**[(**alphabet**.**index**(**char**.**lower**())** **+** shift\_amount**)** **%** **len(**alphabet**)]**

**if** is\_upper**:**

shifted\_char **=** shifted\_char**.**upper**()**

**elif** char**.**isdigit**():**

shift\_amount **=** shift **%** 10

shifted\_char **=** digits**[(**digits**.**index**(**char**)** **+** shift\_amount**)** **%** 10**]**

**elif** char**.**isalnum**()** **and** **not** char**.**isdigit**():**

shift\_amount **=** shift **%** **len(**string**.**punctuation**)** # TODO - !"#$%&'()\*+, -./:;<=>?@[\]^\_`{|}~

shifted\_char **=** string**.**punctuation**[(**string**.**punctuation**.**index**(**char**)** **+** shift\_amount**)** **%** **len(**string**.**punctuation**)]**

**else:**

shifted\_char **=** char

result **+=** shifted\_char

**return** result

**def** caesar\_decipher**(**ciphertext**,** shift**):**

result **=** ""

rus\_alphabet **=** 'абвгдеёжзийклмнопрстуфхцчшщъыьэюя'

lat\_alphabet **=** 'abcdefghijklmnopqrstuvwxyz'

digits **=** '0123456789'

**for** char **in** ciphertext**:**

**if** char**.**isalpha**():**

alphabet **=** lat\_alphabet **if** char**.**lower**()** **in** lat\_alphabet **else** rus\_alphabet

shift\_amount **=** shift **%** **len(**alphabet**)**

is\_upper **=** char**.**isupper**()**

shifted\_char **=** alphabet**[(**alphabet**.**index**(**char**.**lower**())** **-** shift\_amount**)** **%** **len(**alphabet**)]**

**if** is\_upper**:**

shifted\_char **=** shifted\_char**.**upper**()**

**elif** char**.**isdigit**():**

shift\_amount **=** shift **%** 10

shifted\_char **=** digits**[(**digits**.**index**(**char**)** **-** shift\_amount**)** **%** 10**]**

**elif** char**.**isalnum**()** **and** **not** char**.**isdigit**():**

shift\_amount **=** shift **%** **len(**string**.**punctuation**)**

shifted\_char **=** string**.**punctuation**[(**string**.**punctuation**.**index**(**char**)** **-** shift\_amount**)** **%** **len(**string**.**punctuation**)]**

**else:**

shifted\_char **=** char

result **+=** shifted\_char

**return** result

**def** send\_message**(**chat\_displays**,** user1\_name**,** user2\_name**,** encrypted\_message**,** user\_secret\_keys**):**

**if** user\_secret\_keys**.**get**(**user1\_name**)** **is** **not** **None** **and** user\_secret\_keys**.**get**(**

user2\_name**)** **is** **not** **None** **and** user\_secret\_keys**.**get**(**user1\_name**)** **==** user\_secret\_keys**.**get**(**user2\_name**):**

log\_action**(**f"Пользователь {user2\_name} получил зашифрованное сообщение: {encrypted\_message}"**)**

decrypted\_message **=** caesar\_decipher**(**encrypted\_message**,** user\_secret\_keys**.**get**(**user2\_name**))**

log\_action**(**f"Пользователь {user2\_name} расшифровал зашифрованное сообщение: {decrypted\_message}"**)**

**for** users**,** displays **in** chat\_displays**.**items**():**

**if** user1\_name **in** users **and** user2\_name **in** users**:**

**for** display **in** displays**:**

**if** display**.**winfo\_exists**():**

display**.**config**(**state**=**tk**.**NORMAL**)**

current\_time **=** datetime**.**now**().**strftime**(**"%H:%M:%S"**)**

formatted\_message **=** f"{user1\_name} ({current\_time}): {decrypted\_message}"

display**.**insert**(**tk**.**END**,** f"{formatted\_message}\n"**)**

display**.**config**(**state**=**tk**.**DISABLED**)**

**else:**

log\_action**(**f"Пользователь {user2\_name} не получил сообщение"**)**

pair\_key **=** **(**user1\_name**,** user2\_name**)**

**if** pair\_key **in** chat\_displays**:**

displays **=** chat\_displays**[**pair\_key**]**

**for** display **in** displays**:**

**if** **isinstance(**display**,** tk**.**Text**)** **and** display**.**winfo\_exists**():**

display**.**config**(**state**=**tk**.**NORMAL**)**

current\_time **=** datetime**.**now**().**strftime**(**"%H:%M:%S"**)**

decrypted\_message **=** caesar\_decipher**(**encrypted\_message**,** user\_secret\_keys**.**get**(**user1\_name**))**

formatted\_message **=** f"{user1\_name} ({current\_time}): {decrypted\_message}"

display**.**insert**(**tk**.**END**,** f"{formatted\_message}\n"**)**

display**.**config**(**state**=**tk**.**DISABLED**)**

**def** create\_chat**(**user1\_name**,** user2\_name**,** chat\_messages**,** chat\_displays**,** user\_secret\_keys**):**

chat\_window **=** tk**.**Toplevel**(**root**)**

chat\_window**.**title**(**f"{user1\_name}"**)**

chat\_window**.**geometry**(**"600x600"**)**

chat\_display **=** tk**.**Text**(**chat\_window**,** wrap**=**tk**.**WORD**,** state**=**tk**.**DISABLED**)**

chat\_display**.**pack**(**fill**=**tk**.**BOTH**,** expand**=True)**

chat\_displays**.**setdefault**((**user1\_name**,** user2\_name**),** **[]).**append**(**chat\_display**)**

message\_entry **=** tk**.**Entry**(**chat\_window**)**

message\_entry**.**pack**(**fill**=**tk**.**X**)**

**def** exit\_button**(**user1\_name**,** user2\_name**,** connections**,** user\_secret\_keys**):**

**if** are\_connected**(**user1\_name**,** user2\_name**,** connections**):**

connections**[**user1\_name**].**remove**(**user2\_name**)**

chat\_window**.**destroy**()**

**del** chat\_displays**[**user1\_name**,** user2\_name**]**

**del** user\_secret\_keys**[**user1\_name**]**

log\_action**(**f"Пользователь {user1\_name} разорвал соединение с пользователем {user2\_name}"**)**

**def** send\_message\_and\_display**(**user\_secret\_keys**):**

message **=** message\_entry**.**get**()**

**if** message**:**

current\_time **=** datetime**.**now**().**strftime**(**"%H:%M:%S"**)**

encrypted\_message **=** caesar\_cipher**(**message**,** user\_secret\_keys**.**get**(**user1\_name**))**

log\_action**(**f"Пользователь {user1\_name} отправил зашифрованное сообщение: {encrypted\_message}"**)**

chat\_messages**.**append**((**user1\_name**,** user2\_name**,** encrypted\_message**))**

message\_entry**.**delete**(**0**,** tk**.**END**)**

send\_message**(**chat\_displays**,** user1\_name**,** user2\_name**,** encrypted\_message**,** user\_secret\_keys**)**

send\_button **=** tk**.**Button**(**chat\_window**,** text**=**"Отправить"**,** command**=lambda:** send\_message\_and\_display**(**user\_secret\_keys**))**

**def** check\_connection**(**user1\_name**,** user2\_name**):**

**if** chat\_window**.**winfo\_exists**():**

**try:**

**if** are\_connected**(**user1\_name**,** user2\_name**,** connections**):**

send\_button**.**config**(**state**=**tk**.**NORMAL**)**

**else:**

send\_button**.**config**(**state**=**tk**.**DISABLED**)**

**except** **KeyError:**

send\_button**.**config**(**state**=**tk**.**DISABLED**)**

chat\_window**.**after**(**1000**,** **lambda:** check\_connection**(**user1\_name**,** user2\_name**))**

check\_connection**(**user1\_name**,** user2\_name**)**

send\_button**.**pack**()**

close\_button **=** tk**.**Button**(**chat\_window**,** text**=**"Разорвать соединение"**,**

command**=lambda:** exit\_button**(**user1\_name**,** user2\_name**,** connections**,** user\_secret\_keys**))**

close\_button**.**pack**()**

root **=** tk**.**Tk**()**

root**.**title**(**"Diffie-Hellman"**)**

root**.**geometry**(**"950x400"**)**

users\_filename **=** "users.json"

users **=** load\_users\_from\_file**(**users\_filename**)**

connections **=** **{}**

chat\_messages **=** **[]**

user\_secret\_keys **=** **{}**

chat\_displays **=** **{}**

connection\_windows **=** **{}**

root**.**protocol**(**"WM\_DELETE\_WINDOW"**,** **lambda:** **(**save\_users\_to\_file**(**users\_filename**,** users**),** root**.**destroy**()))**

menu\_bar **=** tk**.**Menu**(**root**)**

root**.**config**(**menu**=**menu\_bar**)**

file\_menu **=** tk**.**Menu**(**menu\_bar**,** tearoff**=**0**)**

menu\_bar**.**add\_cascade**(**label**=**"Файл"**,** menu**=**file\_menu**)**

file\_menu**.**add\_command**(**label**=**"Выход"**,** command**=**root**.**destroy**)**

help\_menu **=** tk**.**Menu**(**menu\_bar**,** tearoff**=**0**)**

menu\_bar**.**add\_cascade**(**label**=**"Справка"**,** menu**=**help\_menu**)**

help\_menu**.**add\_command**(**label**=**"О программе"**,** command**=**show\_about**)**

button\_frame **=** tk**.**Frame**(**root**)**

button\_frame**.**grid**(**row**=**2**,** column**=**0**,** columnspan**=**2**)**

create\_user\_button **=** tk**.**Button**(**button\_frame**,** text**=**"Пользователь"**,**

command**=lambda:** create\_user\_window**(**users**,** connections**,** chat\_messages**,** chat\_displays**,**

user\_secret\_keys**,** connection\_windows**))**

create\_user\_button**.**pack**(**side**=**"left"**,** padx**=**5**)**

show\_user\_list\_button **=** tk**.**Button**(**button\_frame**,** text**=**"Показать список пользователей"**,**

command**=lambda:** show\_user\_list**(**users**))**

show\_user\_list\_button**.**pack**(**side**=**"left"**,** padx**=**5**)**

show\_parameters\_button **=** tk**.**Button**(**button\_frame**,** text**=**"Показать параметры p и g"**,** command**=**show\_DH\_parameters**)**

show\_parameters\_button**.**pack**(**side**=**"left"**,** padx**=**5**)**

log\_text **=** Text**(**root**,** wrap**=**tk**.**WORD**,** width**=**110**,** height**=**20**)**

log\_text**.**grid**(**row**=**8**,** column**=**0**,** columnspan**=**4**,** padx**=**10**,** pady**=**10**)**

save\_logs\_button **=** tk**.**Button**(**button\_frame**,** text**=**"Сохранить логи"**,** command**=**save\_logs\_to\_file**)**

save\_logs\_button**.**pack**(**side**=**"left"**,** padx**=**5**)**

display\_logs\_button **=** tk**.**Button**(**button\_frame**,** text**=**"Показать логи"**,** command**=**display\_logs**)**

display\_logs\_button**.**pack**(**side**=**"left"**,** padx**=**5**)**

root**.**mainloop**()**