

Documentation

Sensitive file sharing application

Contents

1	Introduction	1
2	Installation	2
3	GUI	3
4	Functionality description	7

1 Introduction

The aim of this project created for a university course named Cryptography was to create an application for sensitive file sharing. It is designed to be used by users who are on the same network and want share encrypted files without the user of a server, that means using peer-to-peer communication. The application is written in Python 3.12 with the use of supporting libraries which will be described later in this introduction. The app is accompanied by a graphical user interface. Application is designed to be used on any operating system that supports Python 3.12 and the required libraries.

The required files to run this application are all in this repository. Apart from the `main.py` file all of the code is in the `file_share` directory. The `main.py` file is the entry point of the application and is used to start the GUI. As input for the app there is nothing required apart from a file that a user wants to send. The output is a file that was sent to the user from another user using the app.

The application uses the following libraries:

- `Tkinter` - for the Graphical User Interface
- `Uvicorn` - used for reciever realisation
- `FastAPI` - for the reciever API
- `SQLAlchemy` - for the database implementation
- `Cryptography` - for encryption and decryption of files
- `AIOHTTP` - for communication between the sender and the reciever

2 Installation

Installation is done using the following steps for RPM based system. For APT based systems, replace `dnf` with `apt`. For MacOS use [Homebrew](#) package manager.

```
1     sudo dnf install python3 python3-pip python3-virtualenv
2     python3 -m venv .venv
3     source .venv/bin/activate
4     python3 -m pip install pdm
5     pdm install
```

For running the application you can use the following command:

```
1     python3 main.py
```

3 GUI

The GUI start by running command `python3 main.py` in the virtual enviroment.

3.1 Login screen

When the app is opened, the user is greeted by a login screen. The user can either login with an existing account if you used the app in the past as shown in Figure 1 or create a new one wiht username and password as in Figure 2.

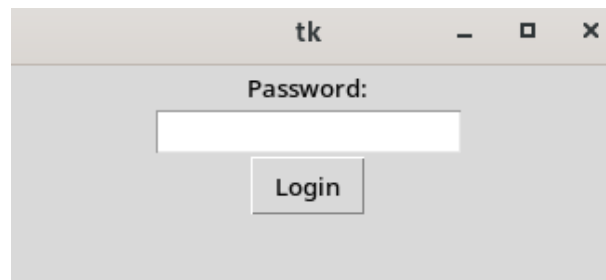


Figure 1: Login screen

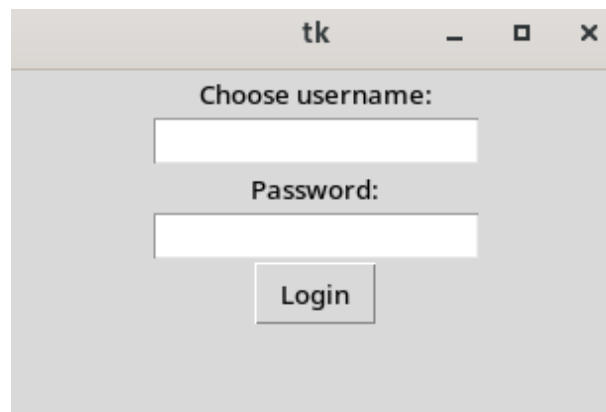


Figure 2: Register screen

3.2 Main window

After the login the user will see the main application window.

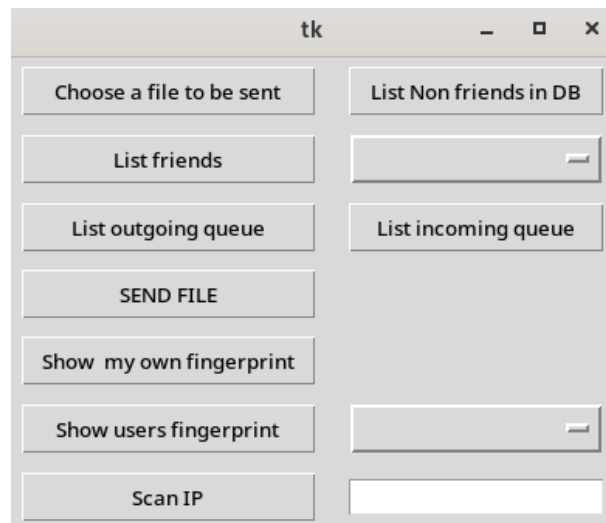


Figure 3: Main window

3.2.1 File selection

Selecting file for transfer is done by the button `Choose a file to be sent` that is shown in Figure 3. After pressing the button, the file selection window will open and user can select the file they want to send.

3.2.2 User discovery and friend addition

There are 2 ways to discover users using the app. Either user can scan the whole network for the active apps that are running using `List Non friends in DB` button. The other

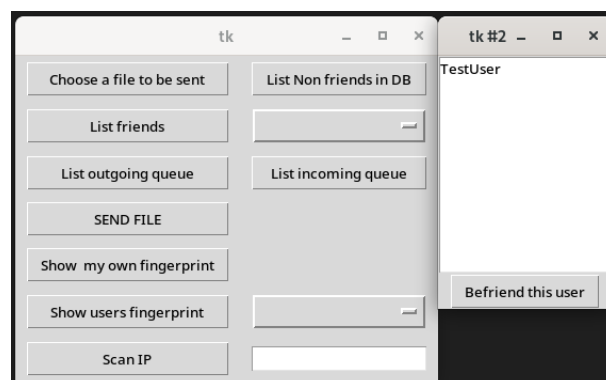


Figure 4: Adding users to db by scanning the network

way is by using button `Scan IP`. User enters the address he wants to scan in the text field next to the button and then presses the button.



Figure 5: Adding user to db by scanning given IP

Adding friends is handled by pressing the button **Befriend this user** that we can see in Figure 4.

3.2.3 Listing of users

There are 2 ways you can list users that user can interact with. Either list only those users who user added to their friend list or list all users in database. For this there are 2 buttons shown on Figure 3. Those are **List friends** and **List Non friends in DB** respectively.

3.2.4 Listing queued files

Files that cannot be sent immediately are queued and will be sent when the user is available. The user can see the list of queued files by pressing the button **List outgoing queue**.

3.2.5 Friend selection

Choosing a friend who will be the target of the file transfer is done by choosing one friend from the rolldown menu next to the **List friends** button.

3.2.6 File sharing

After selecting the file and the friend to send it to, the user can press the button **Send file** to send the file to the selected friend. If the target is not available, the app will prompt the user with following message and put the file in the queue to be sent later.

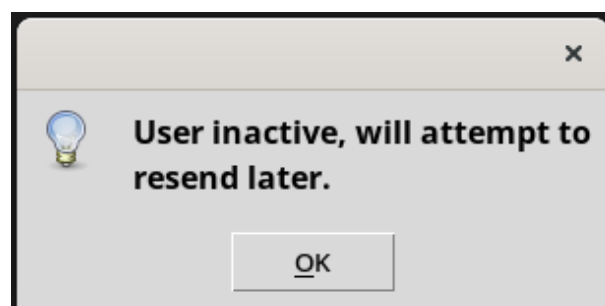


Figure 6: Target not available

Receiving the file is done by the target user pressing the button `List incoming queue` which opens window with all files that are waiting to be received. The user can either save or ignore a single file or accept all files in the queue.

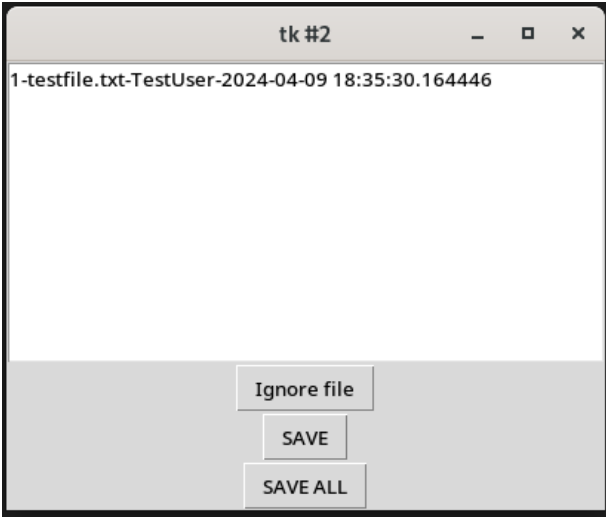


Figure 7: Incoming queue

If the user chooses to save the file he is prompted with window to select the location where the file will be saved.

4 Functionality description