SecurityBox

Security Cloud File Storage System

Team Member

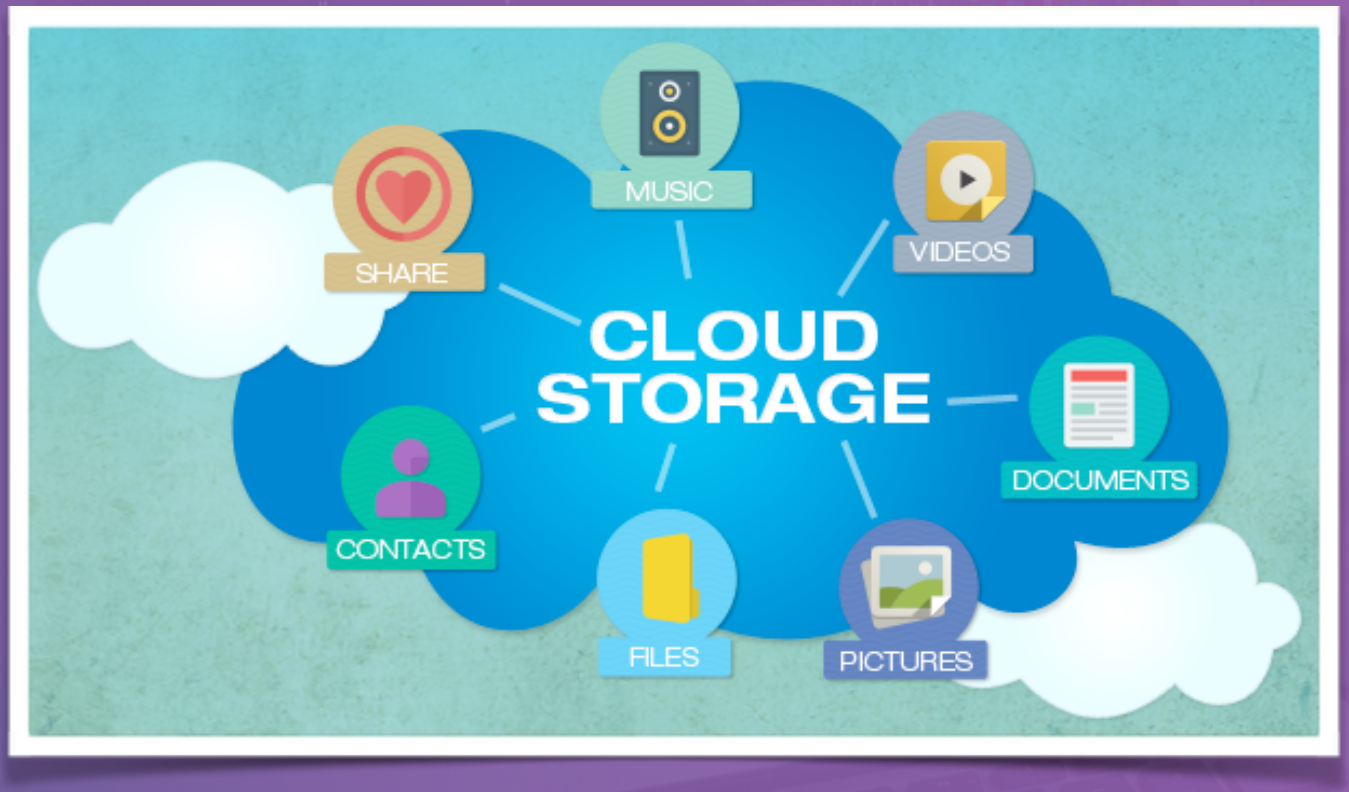
- **Zhaonan Li** (NID: N10838627)
  - Analyze security issues about data eavesdropping, data modification,
    - and find out solutions.
  - Implement AES(CBC) generate key, iv, and encryption function,
    - for encrypting original user files.
  - Implement and design Dropbox connection mechanism,
    - for uploading and downloading encrypted file and HMAC to and from cloud.
  - Implement and design UI functionality, includes:
    - file browser dialog
    - upload and download function.

- **Zinan Liu** (NID: N17115030)
  - Analyze security issues about data eavesdropping and data replay,
    - and find out solutions.
  - Implement AES(CBC) decryption, HMAC verify function,
    - for decrypting encrypted files and verify data integrity.
  - Implement and design retrieve cloud folder structure function.
  - Implement and design UI functionality, includes:
    - list files
    - error logging

- **Jianchen Li** (NID: N10262556)
  - Analyze security issues about data modification.
  - Implement generating HMAC function,
    - for making HMAC of encrypted file for later verification.
  - Implement and design UI functionality, includes:

data processing application

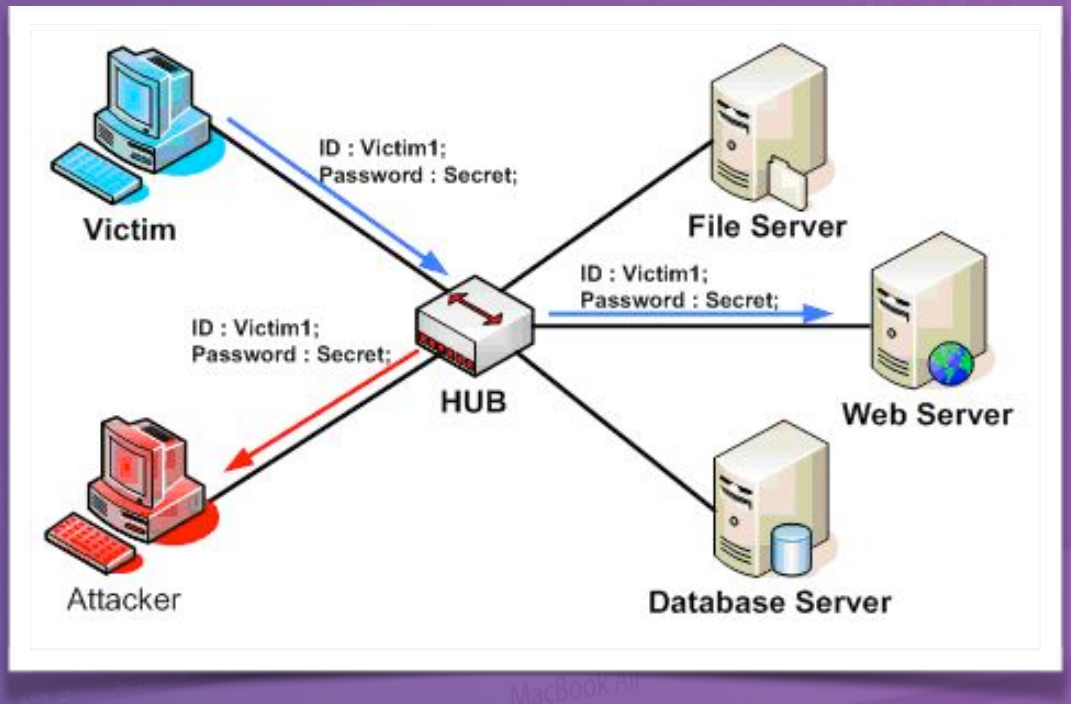
- User stores on and later retrieve files from a cloud server



security issues

- Data eavesdropping

  Files may include information that people don't want to share with other people (e.g. final exam paper).

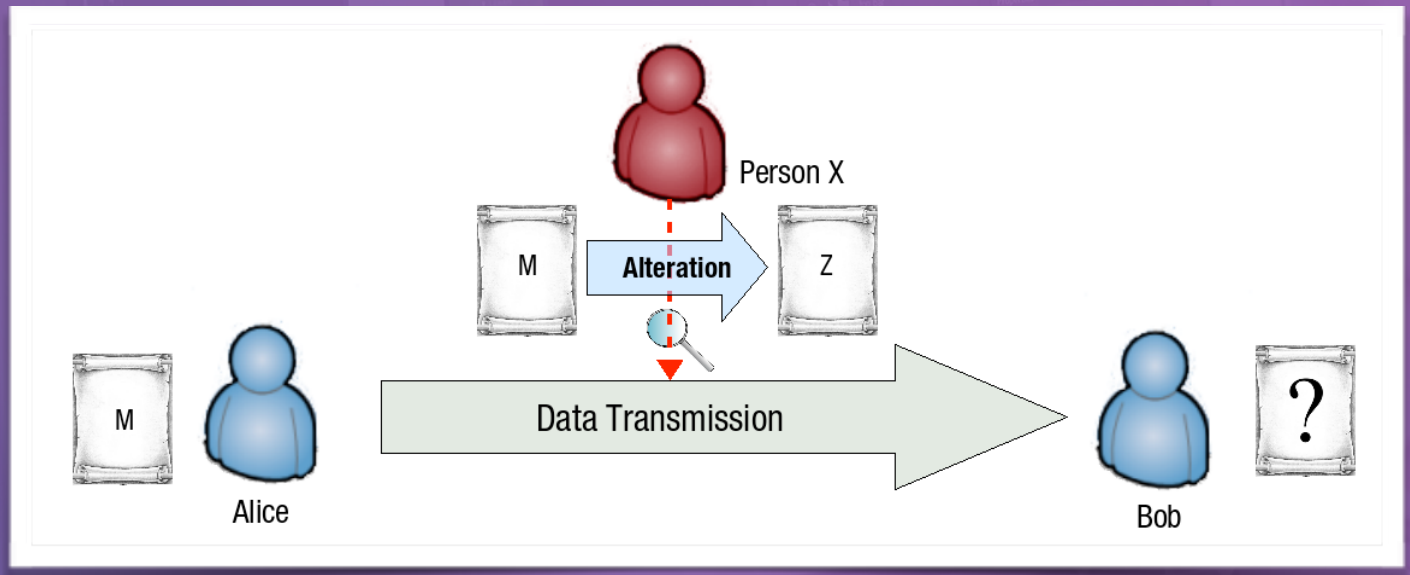


  We can not guarantee every node between user host and server is secure.

security issues

- Data modification

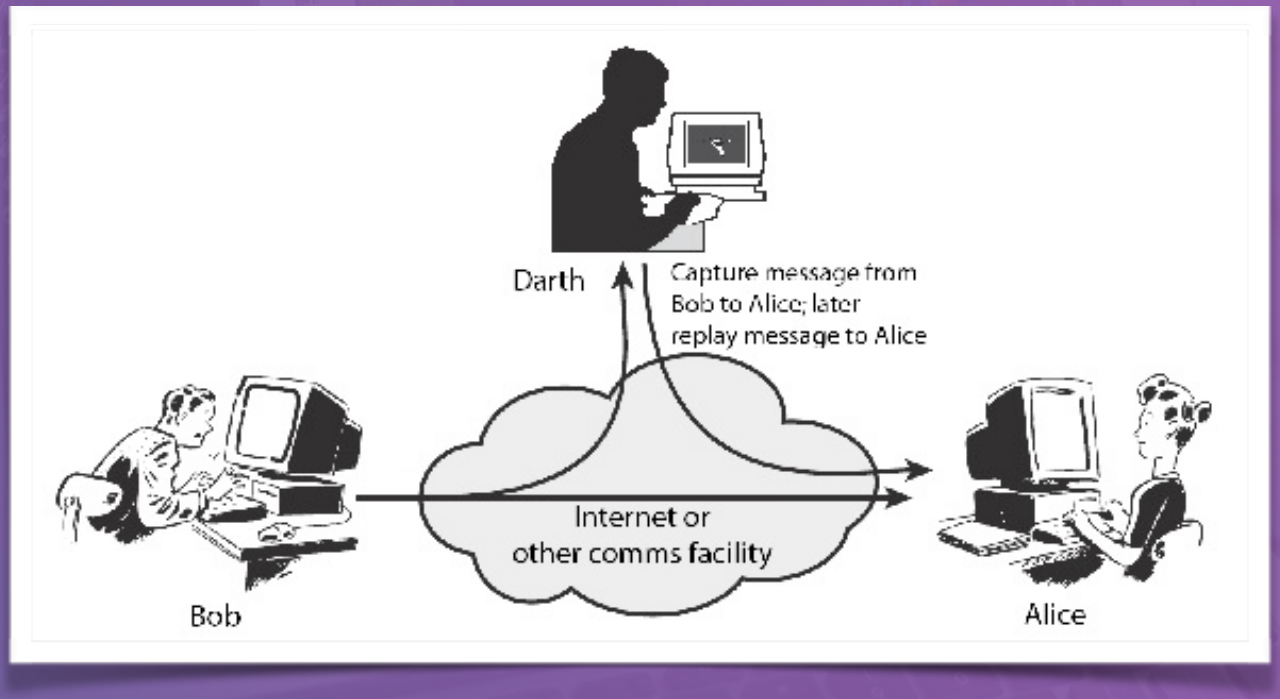
  For example, user stores a program source code on server, and attacker inject some malicious code in it, which may cause serious problems.



security issues

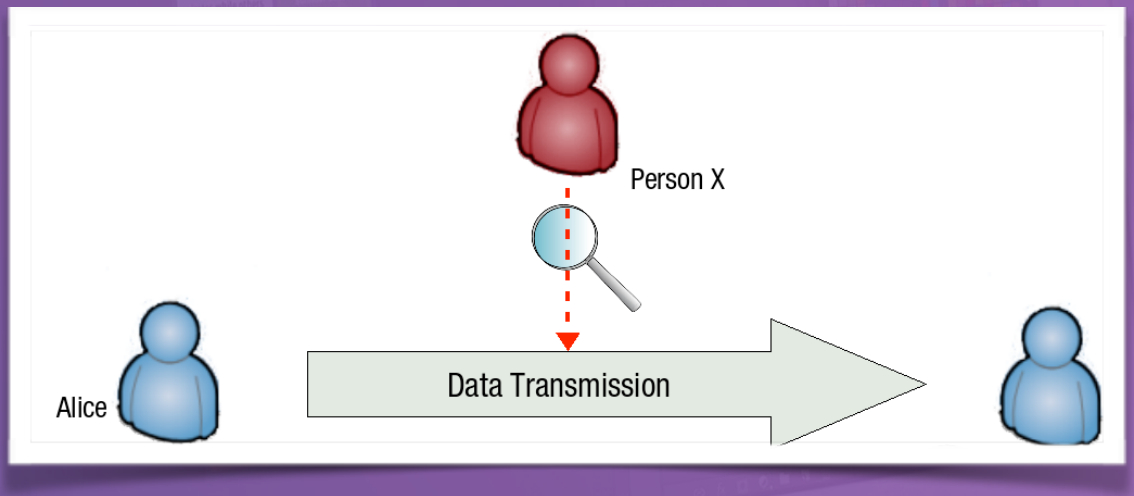
- **Data replay**

Messages between user and server may be sniffed by attacker. Attacker may retransmit the message later without being detected since the message is valid data.



Solutions

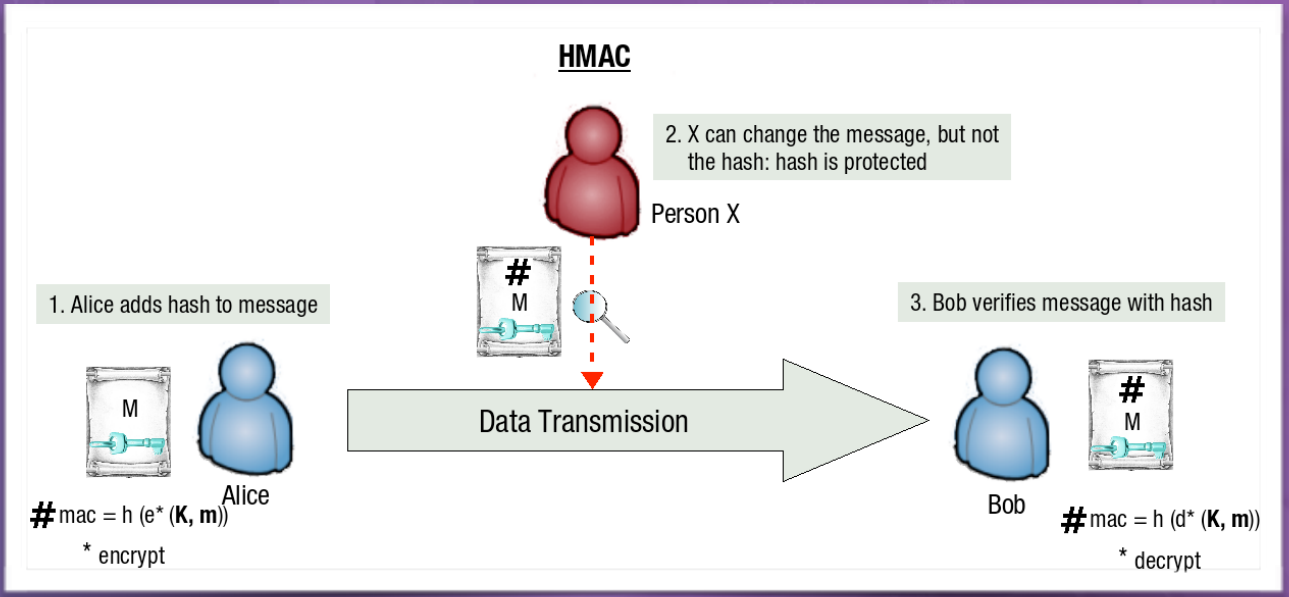
- **Data eavesdropping**
- In order to prevent information disclosure, we encrypt the file before sending it over the internet.
- User keep the key so that only himself can read the data.



- There are several encryption schemes to choose. Consider both efficiency and security, we apply AES CBC mode here for message encryption.

Solutions

- **Data modification**
- Here we use Hash-based message authentication code(HMAC) to verify data integrity.
- User keep the key, so if an attacker intercepts and modifies data, creates a hash, there is no way it will come out correctly.



- We choose HMAC-SHA256 to create HMAC.

Solutions

- **Data replay**
- Use timestamp to prevent replay attack.
- For each packet the user received, we will check the timestamp and compare it with local time. if the difference window between these two values is enough small, user can accept it, otherwise drop it.

Implement

- **Upload:**

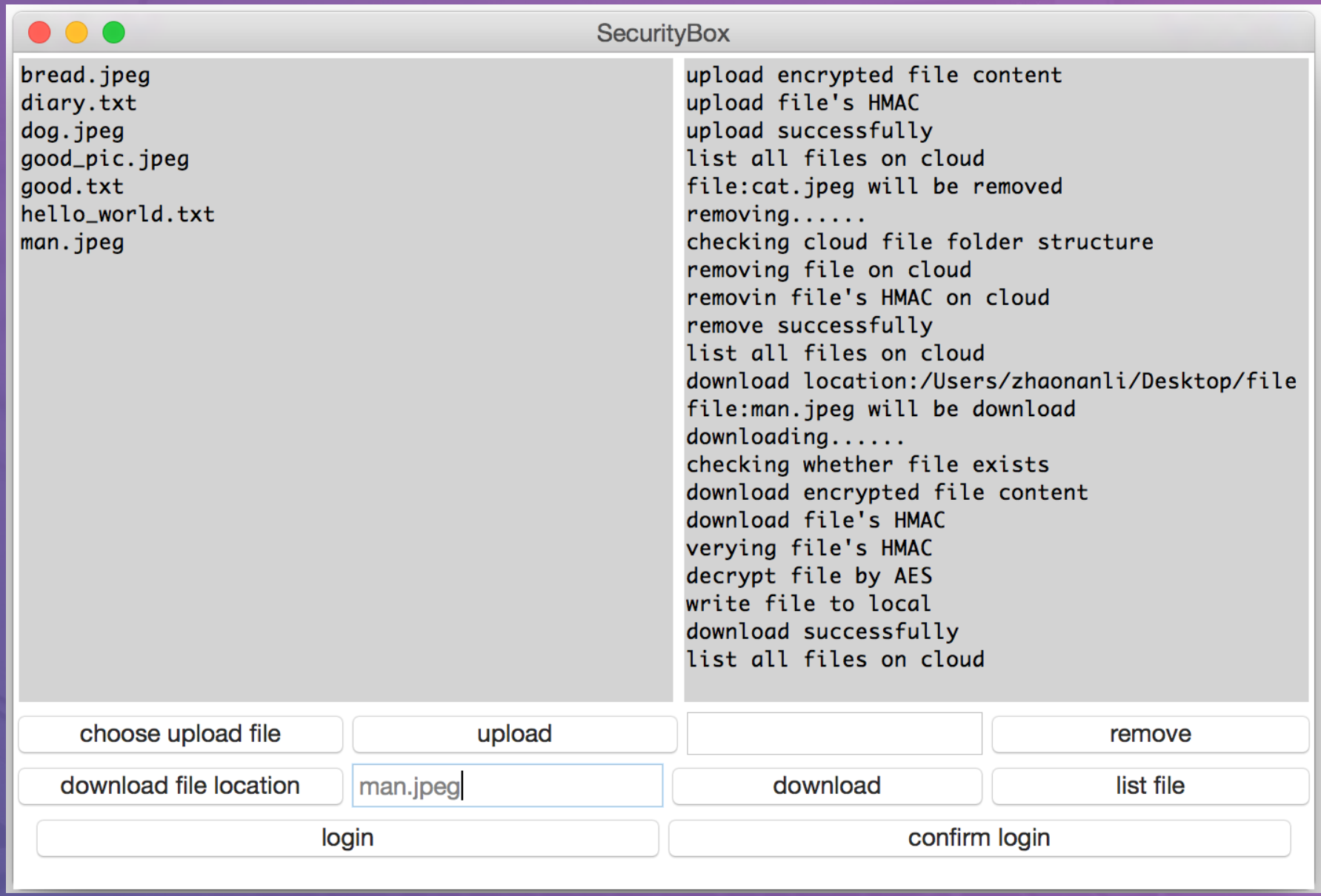
- step 1:Client uses AES-CBC encrypting the file.
- step 2:Client calculate the HMAC of the encrypted file.
- step 3:Client upload file attached with HMAC to cloud server.

Implement

- **Download:**

- step 1:Client check timestamp to decide if accept this packet.
- step 2:Client check HMAC to detect alterations in data.
- step 3:Client decrypt the data and get the original file.

Program Overview



Compile and Run Program

- OS Compatibility: Mac OS X (>= 10.7)
- Library dependencies (need install them on your system):
  - C++11:
    - Crypto++
  - Python 2.7:
    - Tkinter
    - dropbox (https://www.dropbox.com/developers/core)
- Compile:
  - C++ 11 with Crypto++ to compile:
    - AES.cpp
    - HMAC.cpp
- Run (need Python 2.7):
  - python SecurityBox.py

Program Overview

program connecting to Dropbox cloud server

System:

OS: Mac OS X 10.10

Language:

- C++ 11
- Python 2.7

Library:

- C++
  - Crypto++
  - STL
- Python:
  - dropbox
  - Tkinter

File:

Security Part:

- AES.cpp
- HMAC.cpp
- aes_key.txt
- aes_iv.txt
- hmac_key.txt
- SecurityModel.py

Cloud Connect Part:

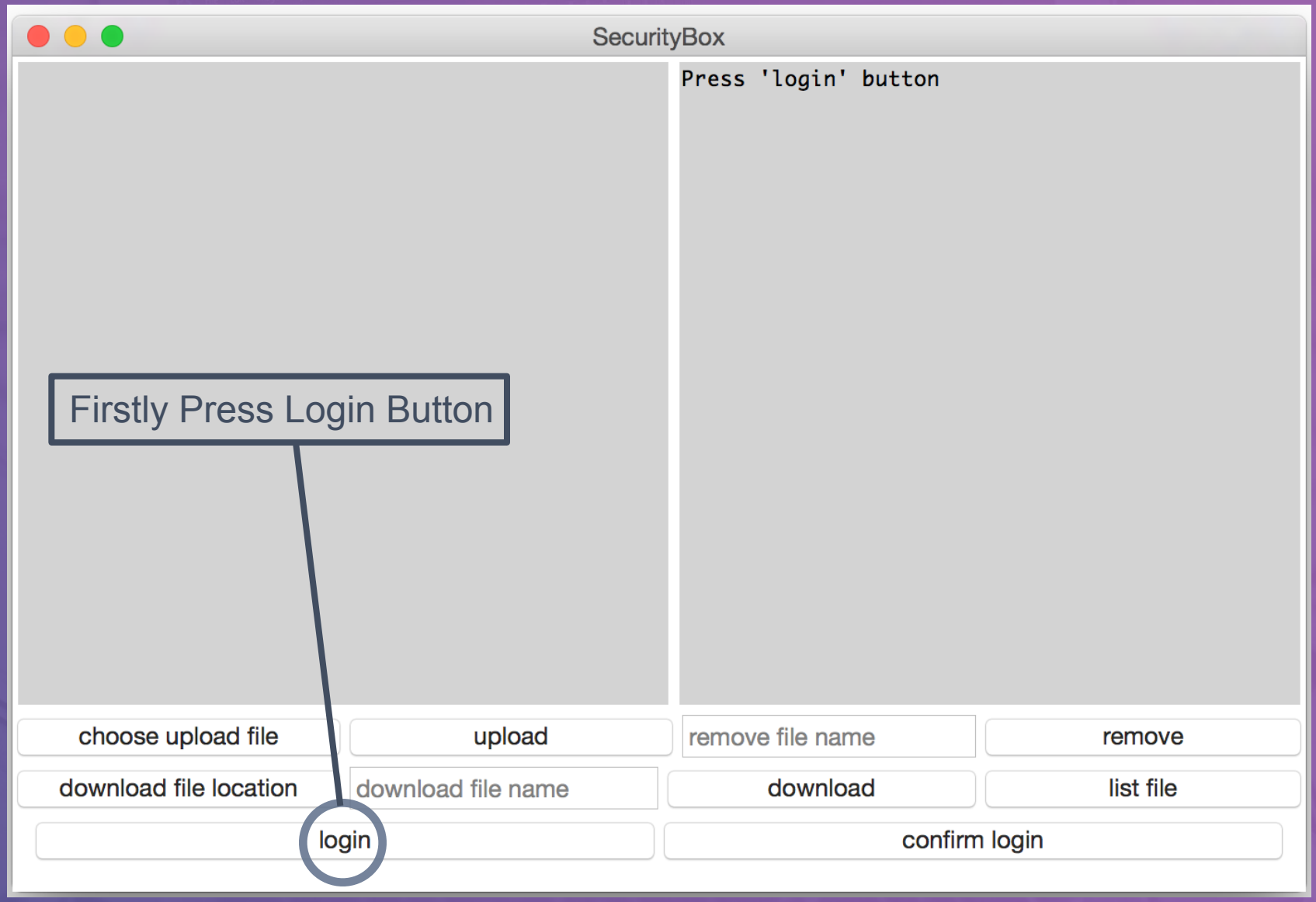
- SecurityBoxModel.py
- SecurityBoxError.py

Main:

- SecurityBox.py

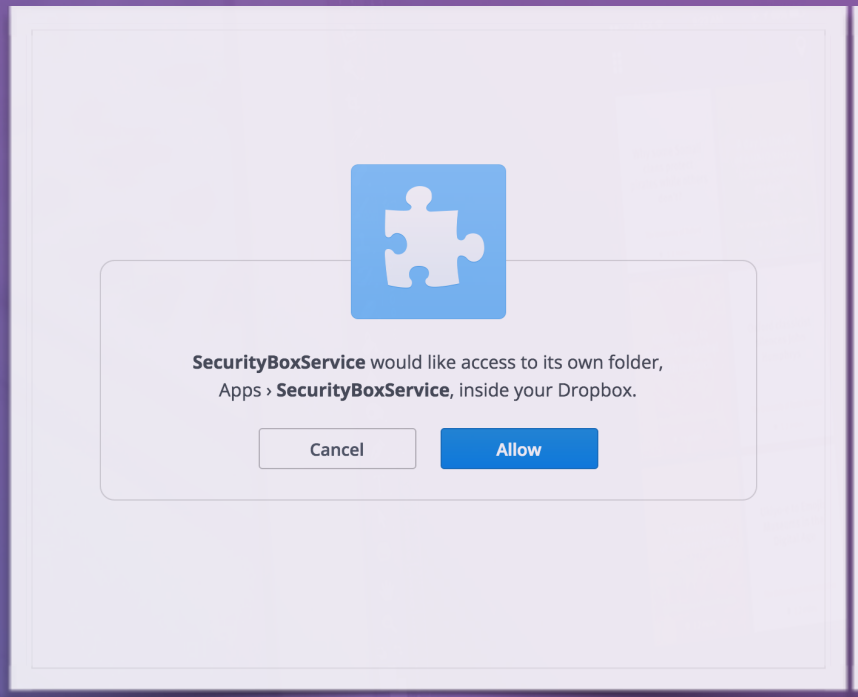
User Login: connect to Dropbox server

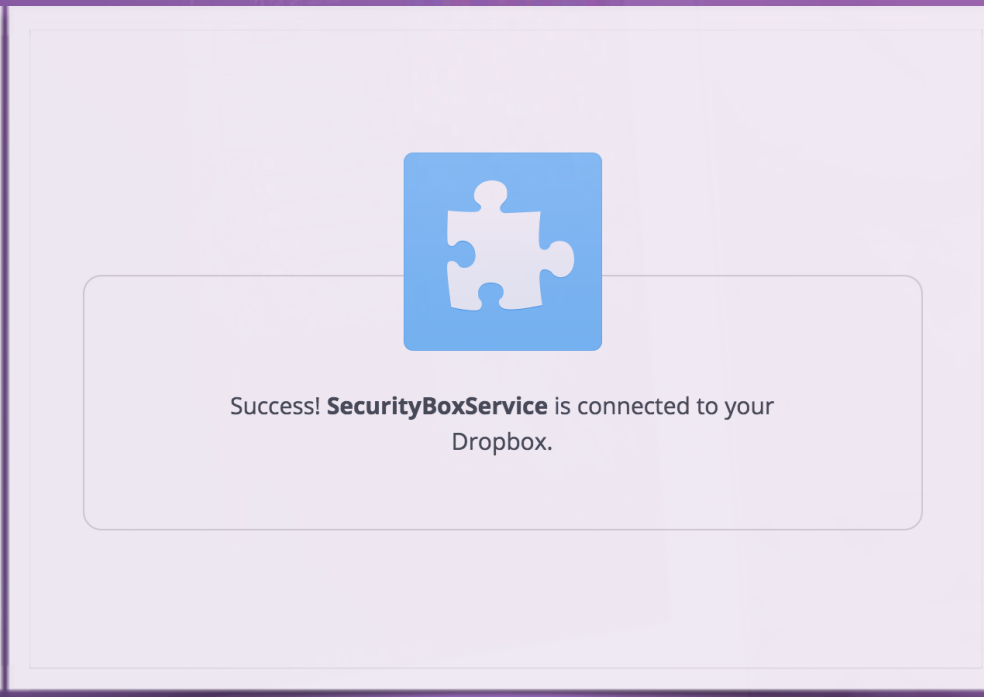
- Firstly, user need login dropbox via OAuth2.0



Program will open web browser automatically

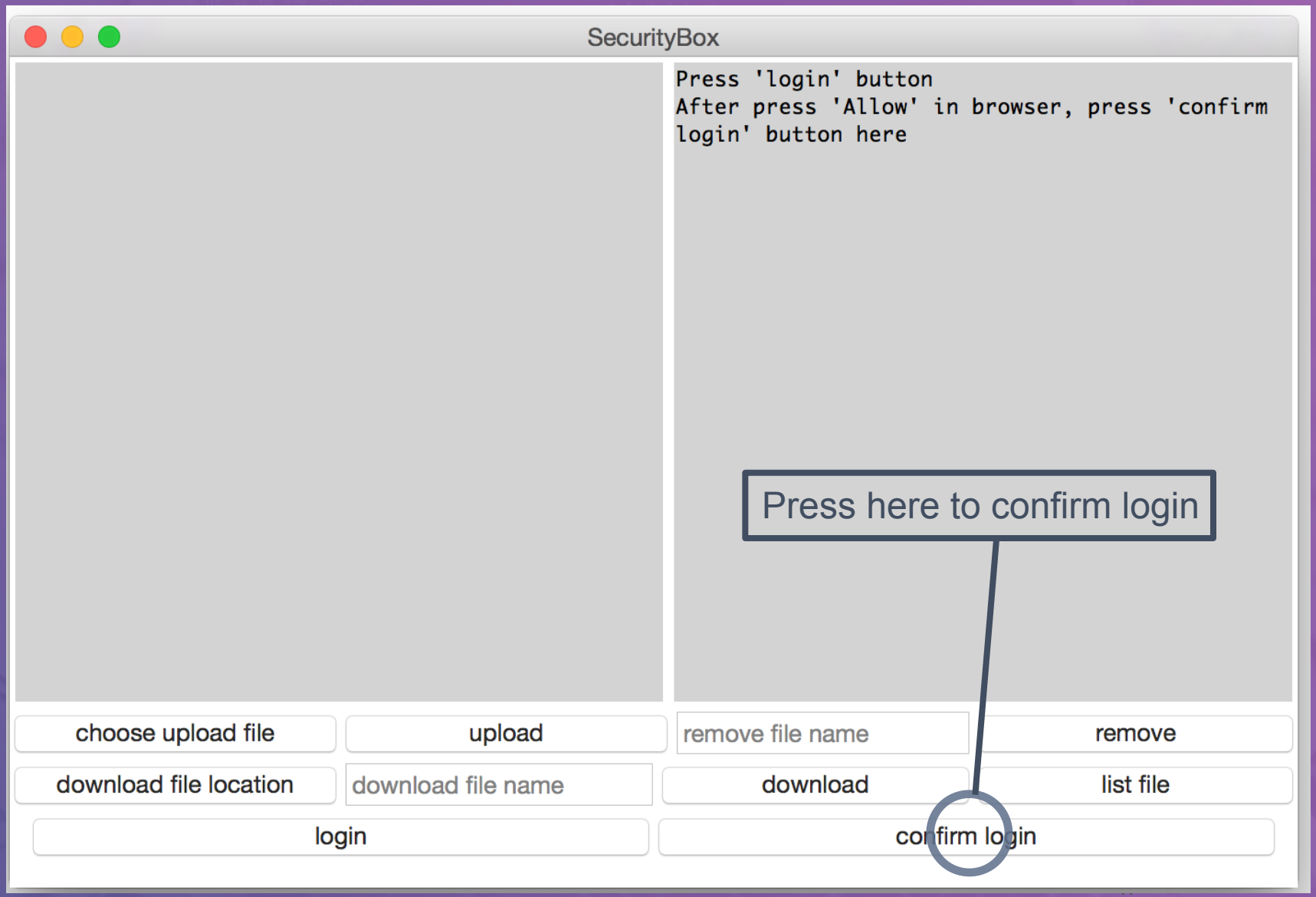
- Press “Allow” to let program login user’s dropbox account via OAuth





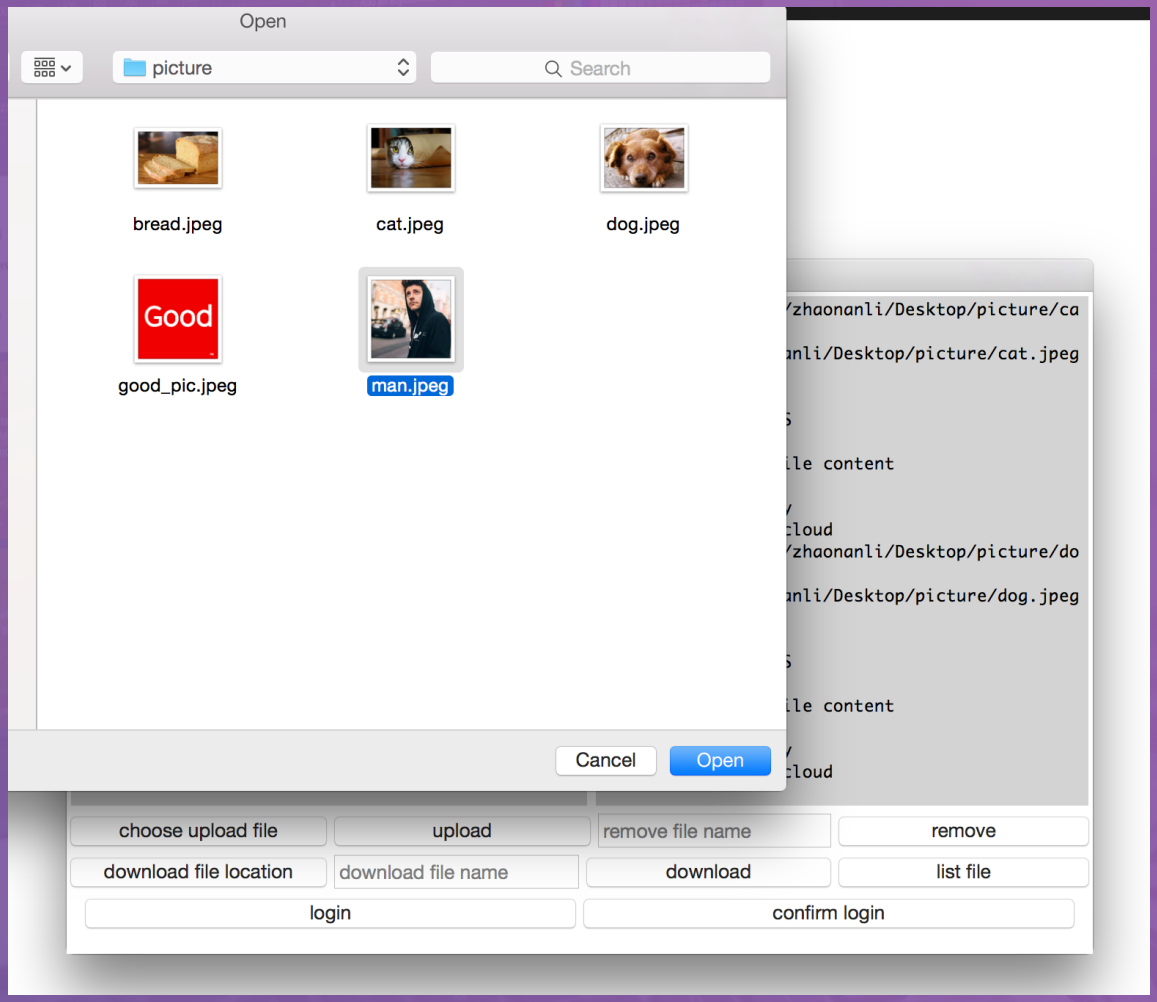
Confirm Login

- After press “Allow” in browser, come back, and press “confirm login” button.

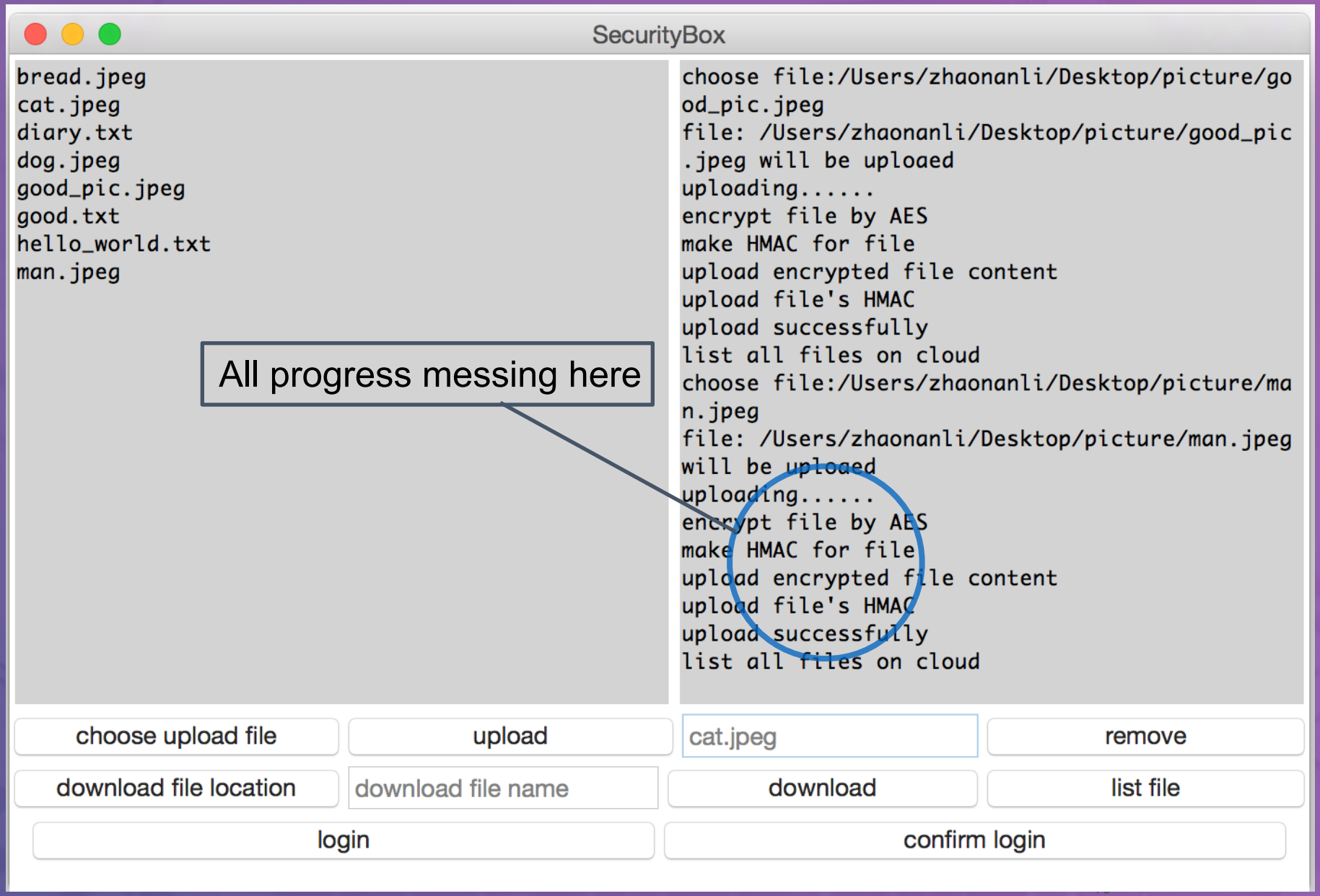


Upload File

- Press “choose upload file” to choose file.
- Then press “upload” button to upload file to cloud.
- Before uploading, program first uses AES(CBC) to encrypt original file, then uses HMAC to generate hmac for encrypted file, finally program uploads encrypted file and file’s hmac to cloud together.

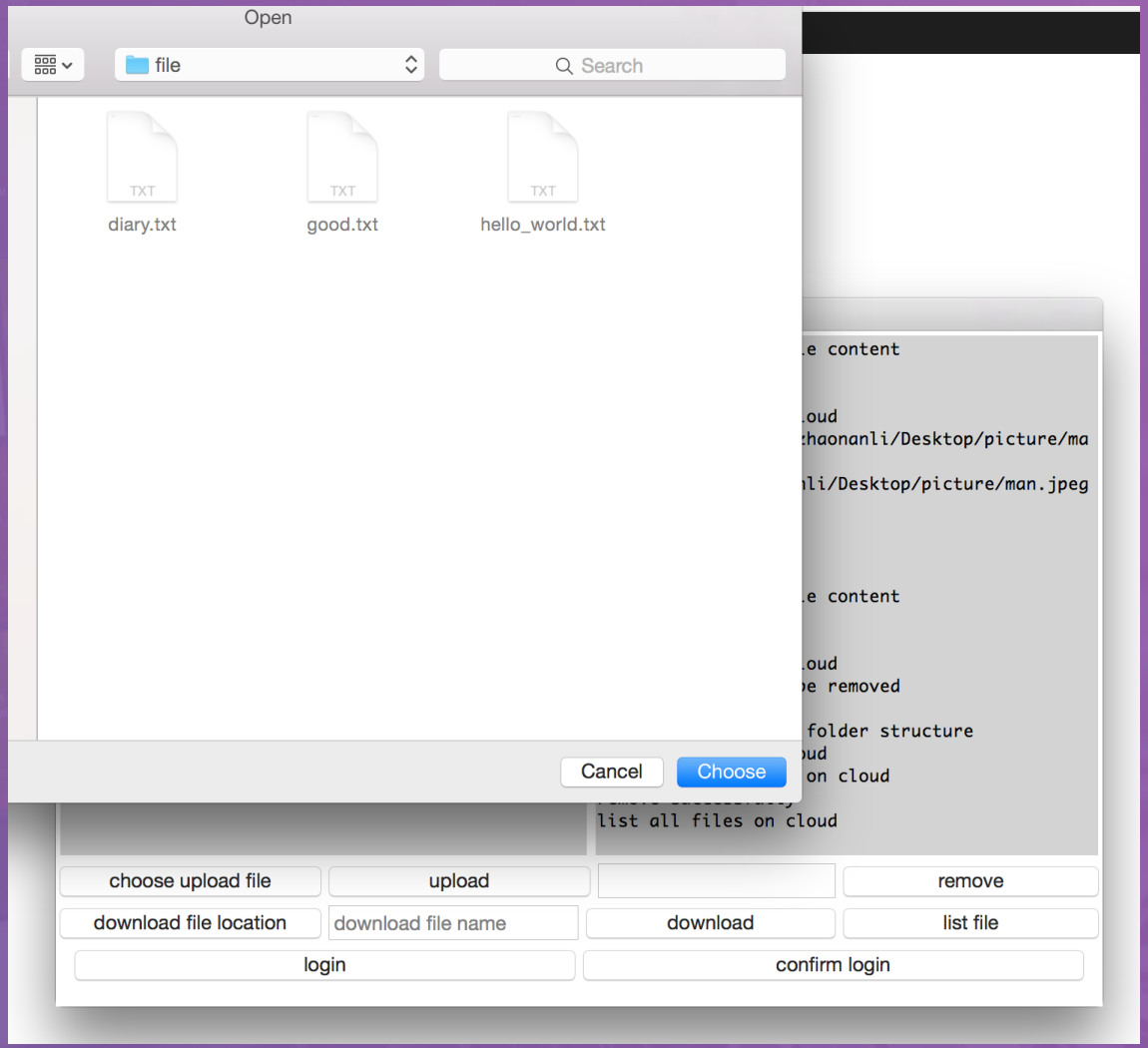


Program shows progress message



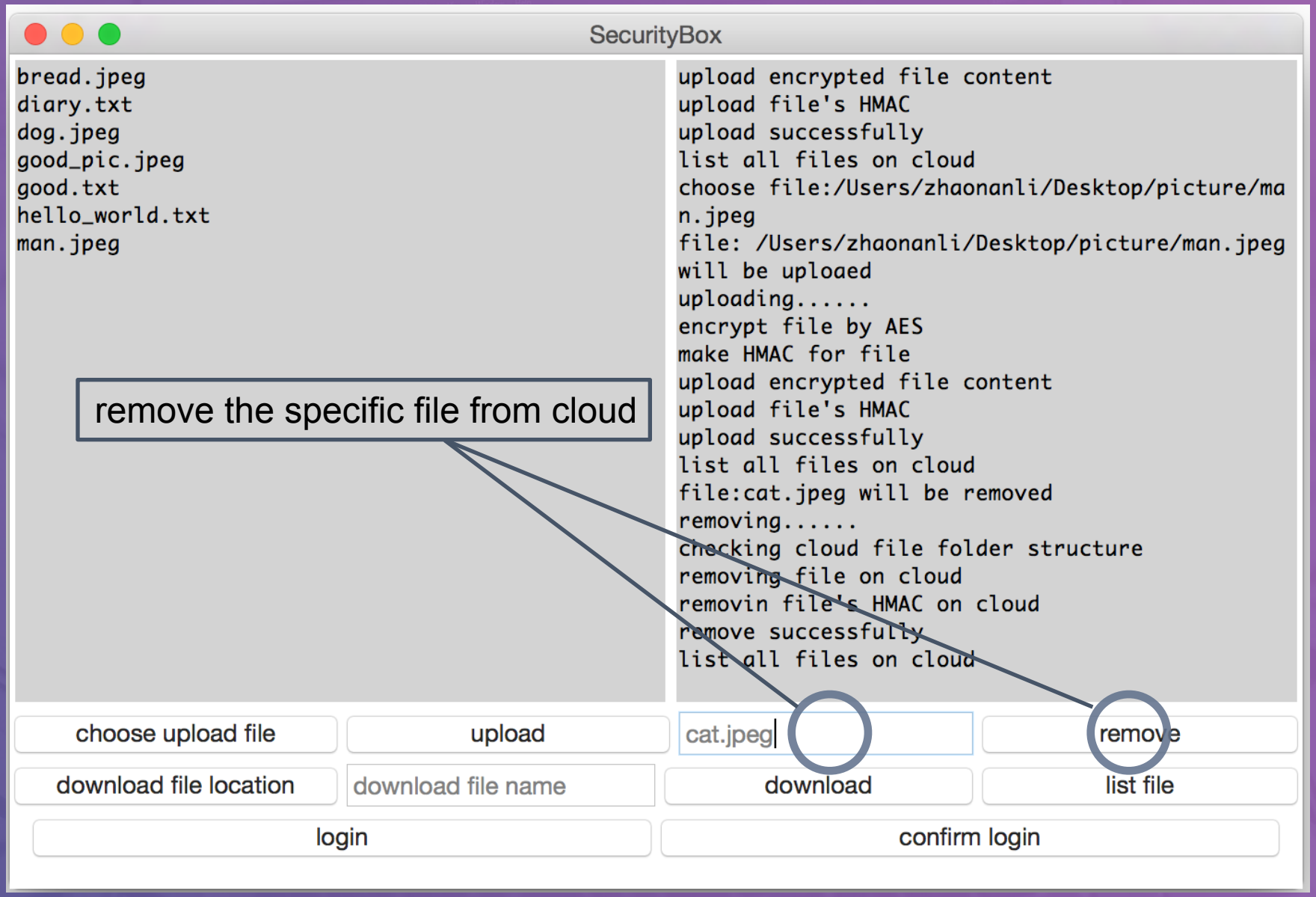
Download file

- Press “download file location” button to choose the destination of downloaded file.
- Then press “download” button to download file from cloud.
- Firstly, program download encrypted file and encrypted file’s hmac from cloud, then program verify hmac, after verification, program uses AES(CBC) to decrypt file, and finally program store decrypted file into local.



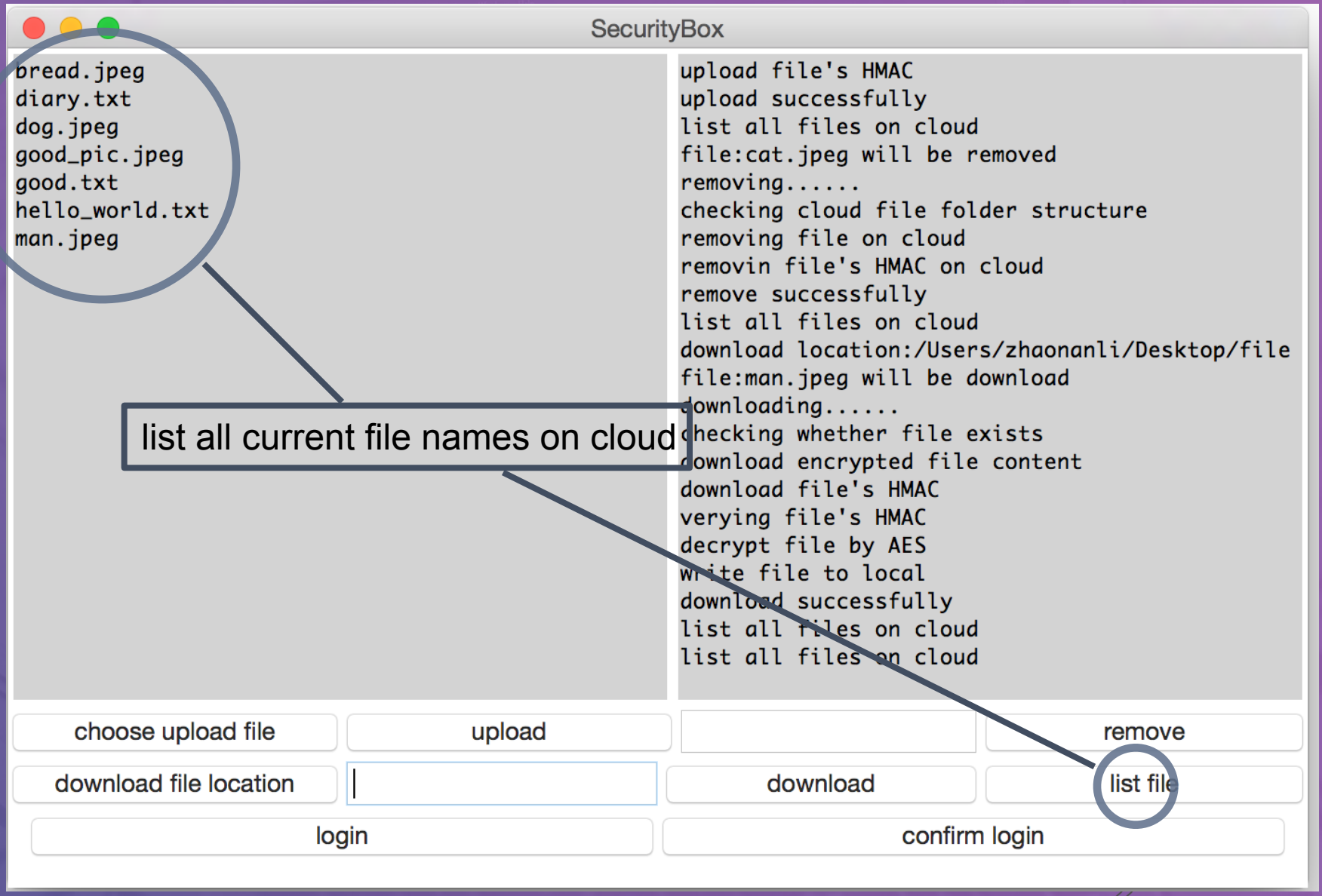
Remove file from cloud

- Enter file name and press “remove” button



List file names on cloud

- press “list file” button to list all file names on cloud



SecurityBox