Multi-user remote password manager

1. General idea

- Client store encrypted file on the server
    - File contains the list of user's password
    - File is encrypted using a key only known by the client (derived from the user's password) (**CIA**)
    - Each user has only one encrypted file on the server, identified by its hashed username
- Client **authenticate** itself to the server using his password
    - Server authenticate itself to the client with its certificate
- Server know which user own which encrypted files on the server but cannot decrypt them
    - Setup **access control** on files:
        - role user : read/write on owned files
        - role admin : add and delete users
- All activities on the server are **logged**
- Transmission between client and server are made on HTTPS (Simulated)

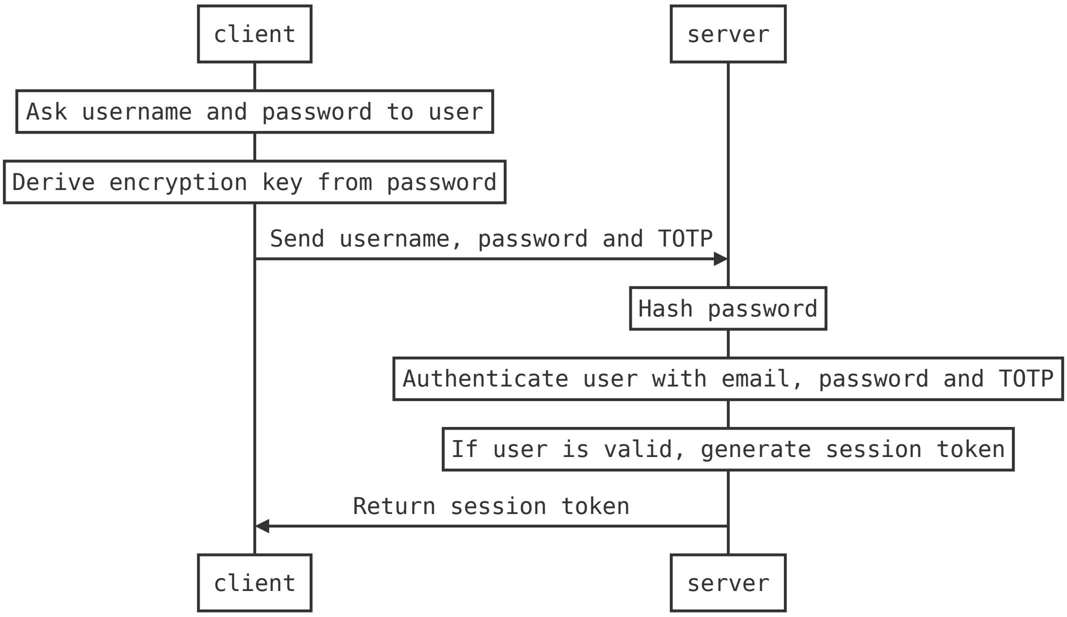
Bonus ? :)

- User use 2FA authenticator to generate a Time-based One Time Password (TOTP).
- Trying to mitigate timing attack by making (sort of) time-constant server endpoint
- Using Diesel for DB management

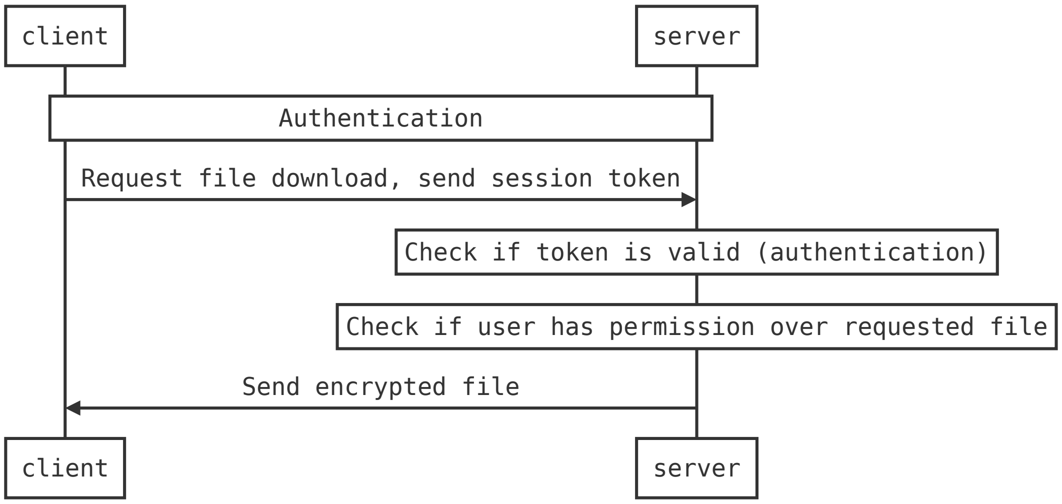
2. Interactions between client and server

Before accessing the password file, the client has to authenticate itself to the server. If validated, he receive a session token that he can use to download and upload the encrypted file.

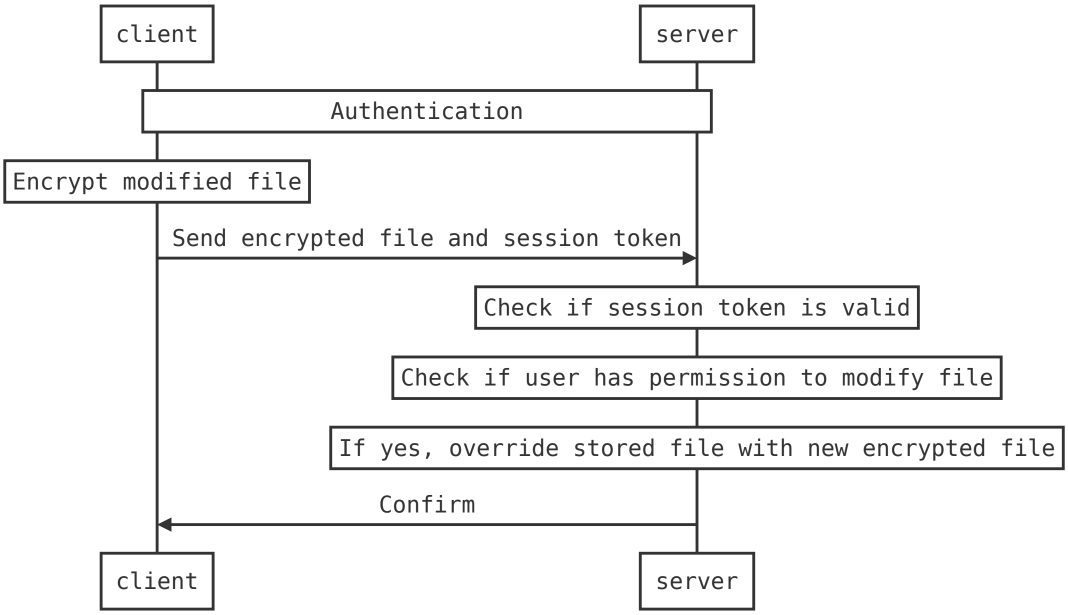
2.1. Authentication


client
server
Ask username and password to user
Derive encryption key from password
Send username, password and TOTP
Hash password
Authenticate user with email, password and TOTP
If user is valid, generate session token
Return session token
client
server

2.2. Download file


client
server
Authentication
Request file download, send session token
Check if token is valid (authentication)
Check if user has permission over requested file
Send encrypted file
client
server

2.3. Upload modified file


client
server
Authentication
Encrypt modified file
Send encrypted file and session token
Check if session token is valid
Check if user has permission to modify file
If yes, override stored file with new encrypted file
Confirm
client
server

3. Interaction between user and client

Once the client is authenticated and has downloaded user's file, user can :

- Read password
- Add new password
- Modify password
- Delete password

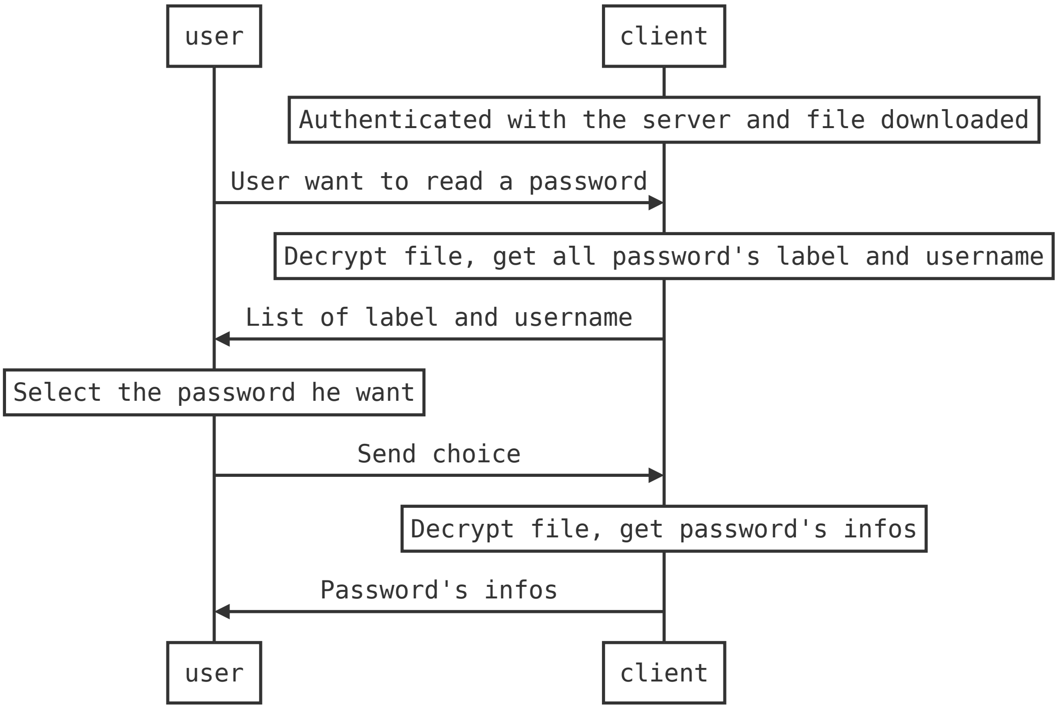
For each option, the encrypted file is decrypted, read/modified, and re-encrypted directly to avoid full decrypted file leak from memory.

For the last 3 options, the client update the passwords' file, encrypt it and send it to the server. The server override the old file with the new encrypted file.

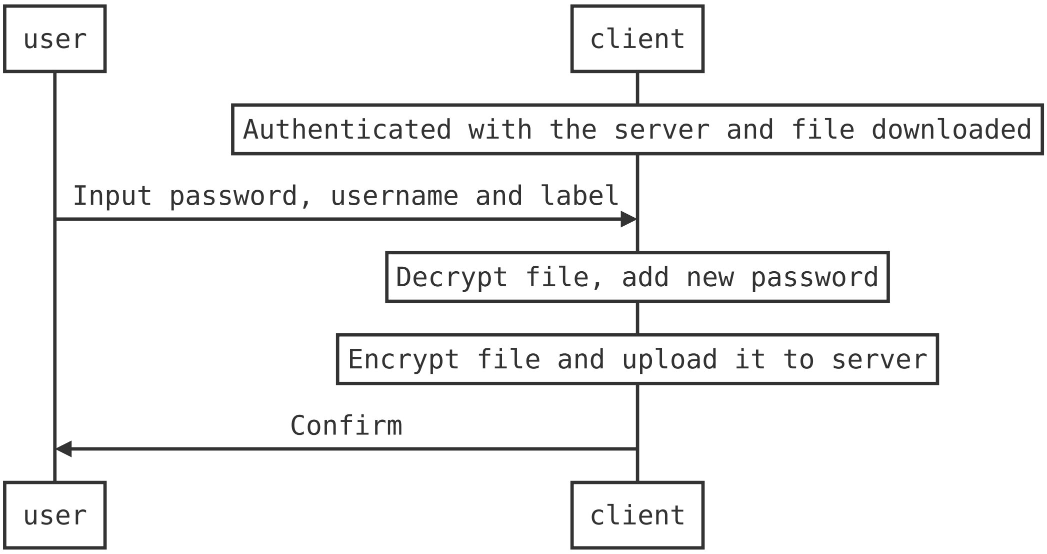
If user has role admin, he can also :

- Add user
- Delete user

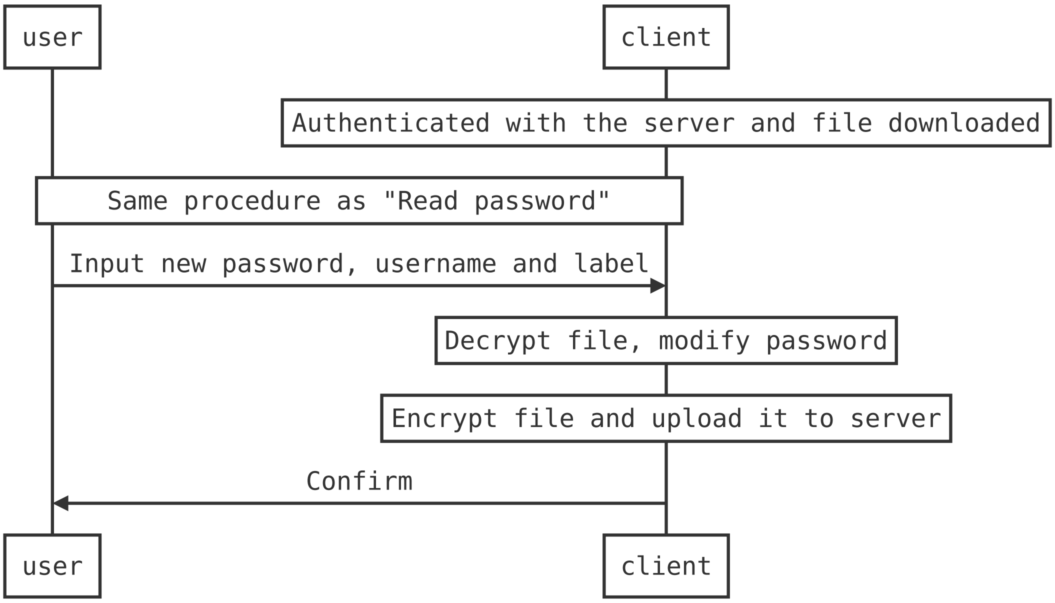
3.1. Read password


user
client
Authenticated with the server and file downloaded
User want to read a password
Decrypt file, get all password's label and username
List of label and username
Select the password he want
Send choice
Decrypt file, get password's infos
Password's infos
user
client

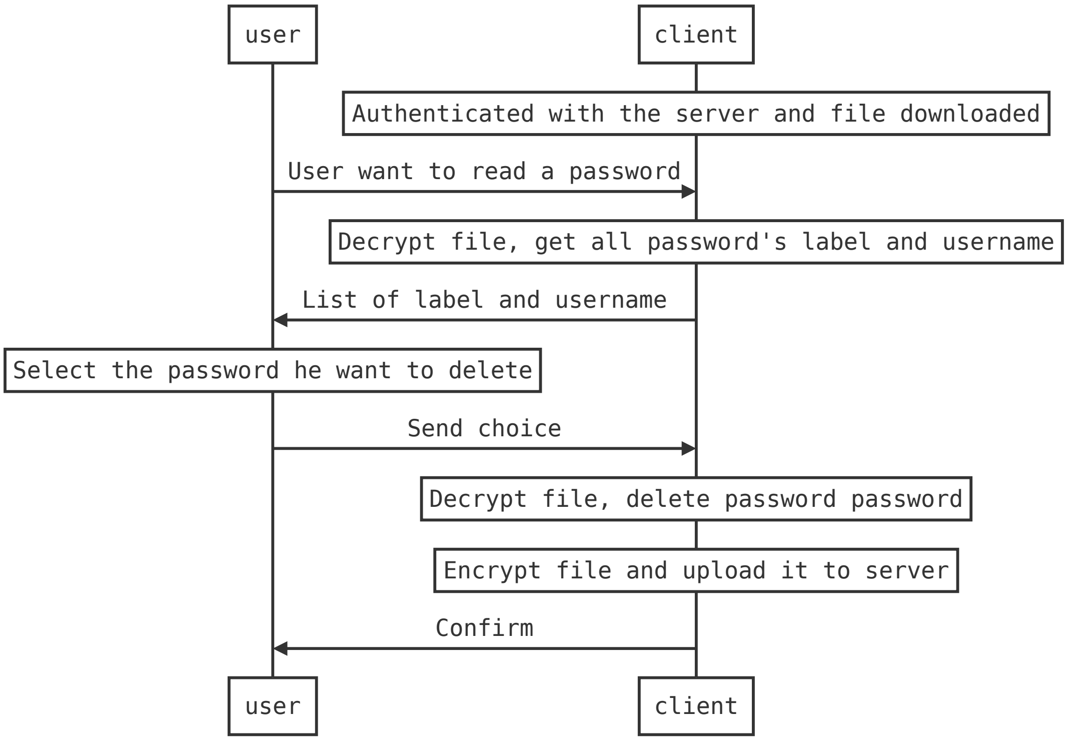
3.2. Add new password


user
client
Authenticated with the server and file downloaded
Input password, username and label
Decrypt file, add new password
Encrypt file and upload it to server
Confirm
user
client

3.3. Modify password


user
client
Authenticated with the server and file downloaded
Same procedure as "Read password"
Input new password, username and label
Decrypt file, modify password
Encrypt file and upload it to server
Confirm
user
client

3.4. Delete password


user
client
Authenticated with the server and file downloaded
User want to read a password
Decrypt file, get all password's label and username
List of label and username
Select the password he want to delete
Send choice
Decrypt file, delete password password
Encrypt file and upload it to server
Confirm
user
client