Do Man Done - Oprah Secure Personal Cloud

Rajdeep Yadav	Poorvi Hebbar	Ramolla Nikhil Reddy
170050034	170050094	170050096
rajdeepyadav004@gmail.com	poorvigpo@gmail.com	rnr1410@gmail.com

Declaration

I acknowledge and understand that plagiarism is wrong. This project is our group's own work. I acknowledge that copying some-one else's work, or part of it, is wrong, and that submitting identical work to others constitutes a form of plagiarism.

1 Introduction

In this project, we are implementing a cloud based system where the user has complete control over encryption schema of data that is uploaded to the server.We have implemented clients on two different platforms.

- **Server**: A completely secure environment for users to store files and folders and is capable of managing multiple clients, and allows registering for new users and logging in for the already registered users.
- **Linux Client**: A client which interacts with spc system from a terminal. A already registered user can not only access the data, he or she had stored in the cloud as a user, but also, create, make changes and then sync the data stored in the cloud with that in his pc, in whichever he way he wants.
- **Web Client**: He can view the contents as well as upload download files folders from the cloud.

All that is saved in the server is encrypted data and is decrypted whenever it is to be accessed. The user can choose any of the 3 encryption schema options which are available.

2 Motivation

Cloud Storage is a service where data is remotely maintained, managed, and backed up. The project has a wide range of practical usages.

- **Usability**: Users can upload and Download files between the cloud storage and their local storage.
- **Accessibility**: Stored files can be accessed from anywhere via Internet connection.
- **Disaster Recovery**: Cloud storage can be used as a back-up plan.
- **Security**: Every user has a key which is required to access the his files.Also the files stored in the server are encrypted with the encryption schema being chosen by the user himself.

Apart from these, learning NodeJS, encryption schemes, socket programming, SQLFS Database management systems was also a major motivation behind the project.

3 General Features

3.1 Web Server

The server manages several clients and manages a database to create and delete folders and files of different users. It allows new users to register, and registered users to login, access and update existing files,as well as upload and download files and folders. The main tools used for the server are node.js, passport.js, express.js, handlebars(for the view engine). SQL-FS database management was also used.

3.2 Web Client

The web client has the ability to access the stored data, upload, download and create new folders and files once regisered and logged in.

- **Register & Login**: One can register using a username and password, and once registered he can login from anywhere using a secure web connection.
- **Upload and Download**: A registered user can download as well as upload any file, folder in the cloud, after chosing the proper encryption schema and giving the proper key.

Some of its other functionalities include:

- **Web Design**: The web design is very user friendly and all the folders and files can be viewed, deleted or downloaded. The folders are links to the list of files of and folders contained within the parent folder.
- We used jQuerry, Boottstrap 4.1.1 and sweetalert for the web-design.
- A user has presently 2 available options for the encryption and decryption schemes : AES_CBC, AES_CTR.

3.3 Linux Client

The linux client accesses the data stored in the server using the terminal.

- **Initialisation**: Given a username and his preferences, a unique key is generated.
- **Configuration**: The user can configure his credentials, and that would be updated in the database of the server.
- **Sync**: As a pc is being used, the sync between the client and the server can be done in 6 different ways. A Brief description of them is as follows:
  - Push - Files gets transferred from the client to the server, if not present in the latter one. The user is given the choice, when there is a clash between the contents.
  - Push(forced) -Files get transferred such that the server has the same content as that of the client.
  - Pull - Files gets transferred from the server to the client, if not present in the latter one. The user is given the choice, when there is a clash between the contents.
  - Pull(forced) -Files get transferred such that the client has the same content as that of the server.
  - Ideal-sync - The data of the server as well as the client is updated to that of the union of the entire data in the client as well as the server. It is done by sync pushing and then pulling without force.
  - Periodic Sync - Ideal-sync is established after every 10minutes.

Some of its other functionalities include:

- Uploading (after encryption) and Downloading (after decryption) files and folders, accessing them and also making changes to them as a registered user.-TODO
- A user has presently 2 available options for the encryption and decryption schemes : AES_CBC, AES_CTR.
- The important python packages that were used for the Linux client: PyCrypto and hashlib.

4 Design Choices

- We used **Node.js** instead of Django for developing the server, so that it supports parallelism and is more faster and efficient.
- A user is provided with 2 encryption schema options AES_CBC and AES_CTR, to chose from.
- We give a linux client the choice to sync with the cloud in 5 different ways.
- Periodic sync is also implemented.

References

[1] express https://expressjs.com/

[2] express-generator https://expressjs.com/en/starter/generator.html

[3] passportjs http://www.passportjs.org/

[4] handlebars https://handlebarsjs.com/

[5] nodejs https://nodejs.org/

[6] jquery https://jquery.com/

[7] bootstrap 4.1.1 http://blog.getbootstrap.com/2018/04/30/bootstrap-4-1-1/

[8] sweetalert https://sweetalert2.github.io/

[9] Lots of stackoverflow https://www.stackoverflow.com

[10] Some Mozilla CDN https://developer.mozilla.org/