CMPE 494 - Free Security Project Report

Kayacan Vesek - Adil Numan Çelik

Secure Chat App

Description

In the last few weeks, messaging apps were one of the main topics in social media. So, we wanted to design a secure app for customers, Secure Chat App provides an end-to-end messaging by using asymmetric encryption.

What we aimed at this project:

We wanted to create a secure chat app with these requirements:

- End-to-end encryption
- Only receiver and sender should be able to see contents of a message
- Messages should not be stored in devices, they should only be seen when the user uses the app
-

Requirements to run

- **Client Side:**

Android Operating System, Internet Connection.

- **Server Side:**

Nodejs v10.19.0

Architectural Design

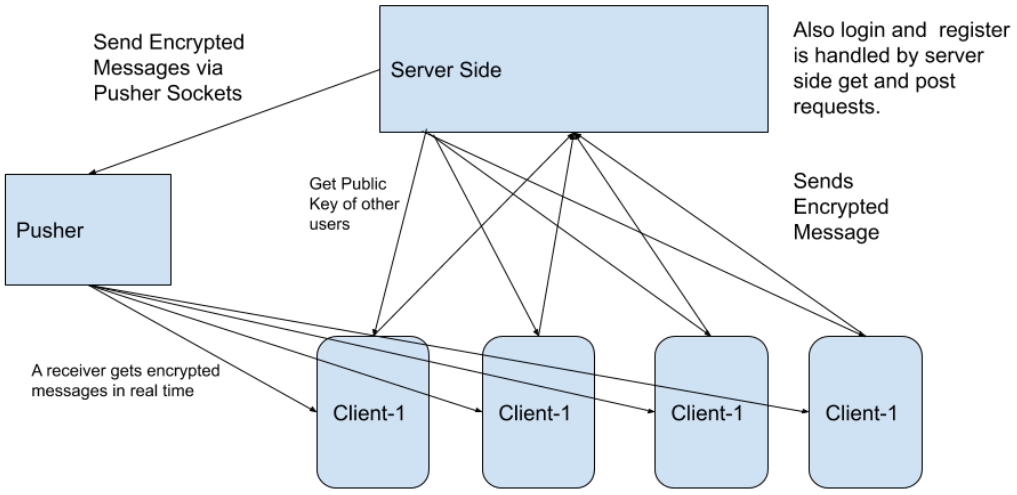
The app consists of two parts: server app and client app.
Server app stores user's information in the Mongo Database. These informations are:

- User id
- User name
- User's socket information
- User's public key

When a client app is opened on a device for the first time it creates a public key and a private key. It stores the private key in the device and sends the public key to the server together with the user's other information. After this, this device is registered on the server.

All messages are encrypted by the receiver's public key so that only the receiver can decrypt them. Even, the server can not access the contents of a message.

Current version of the app only supports user to user messaging when both users are online and they opened the app. A user can not see a message they receive when the app is not open.



Messages are always encrypted(RSA) with receiver client's public key to maintain security.

Development Details

We utilized an api called Pusher. We also utilized a free to use tutorial shared in Pusher's official website. Pusher handles communication between clients. It also helps notifying other clients whether a client is online or not. (?)

Server side app uses node.js, stores information in a MongoDb database and runs on a DigitalOcean server.

Client side app is an Android app that can work on any Android device with Android version 4.1 or newer.

Public and private keys are generated using the RSA algorithm by Java Security Package.

How to install and run the code.

- **Client side:**

Install the app using the apk file given on the repository.

- **Server side**

Install the necessary NPM packages using this command:

```
$ npm install
```

Run MongoDB using this command:

```
$ mongod --dbpath C:\MongoDB\data\db # Windows
$ mongod --dbpath=/path/to/db/directory # Mac or Linux
```

Run the application using this command:

```
$ node index.js
```

The app will be here: http://localhost:5000.