Team FSociety

An End-to-End Encryption Chat Application Overview

Estefany Montoya
Rone Caballero
Prof. Mehrdad Aliasgari
December 6, 2016

This is a Reboot

Application Properties

Before:

- **Java on Spring - hand picked language**
- **AWS - something about buying websites**
- **JSON - Jay who?**
- **MySQL - database**
- **SSL(1.2) - the red tunnel**
- **PGP - green tunnel**

After:

- **Ruby on Rails - language and platform**
- **AWS for SSL, but used Heroku - web service**
- **GoDaddy - Domain name**
- **JSON - Authentication tokens, stateless**
- **SQLite3 - Database**
- **SSL(1.2) - RSA**
- **Qualsys SSL labs - To check RSA**
- **PGP - for message integrity, in progress**
- **Postman - To check functions**
- **C# forms - client**

Assets

- To protect the user's Confidentiality, Integrity, and Availability with the app.
- Message Integrity
- Stakeholders: Users

Ruby on Rails

Pros:

- **Gems = Premade libraries**
- **Less code implementation than Java or C++**

Cons:

- **If you screw up, you have to find your error without any debugging tools**
- **If you have not learned the language before, you’re going to have a bad time**

AWS v Heroku

AWS:

Pros:

- **Free**
- **Can customize SSL for perfect A+ on Qualsys**

Cons:

- **Could not connect with Rails as easily**
- **Lots of configurations to do**

Heroku:

Pros:

- **They practically make the configurations for you**
- **Rails friendly**

Cons:

- **Only solid A in Qualsys labs, cannot customize the SSL configurations**
- **Only free if you are a student**

GoDaddy

Pro:

- **Popular website for domain names**

Cons:

- **Hella expensive**
- **Most domain names are bought**
- **Other websites can provide cheaper deals**

MySQL v SQLite3

Not much difference, mostly preference base. However, if one chooses PHP to code the project, then you can just install LAMP to get the package of: Linux, Apache, MySQL, and PHP.

SQLite3 will be used for all storage purposes such as usernames and passwords, user ids and conversation ids, message bodies.

JavaScript Object Notation Web Token

- **Stateless meaning the server doesn’t have to keep checking the user if they are authenticated users.**
- **No logouts**

Utility helpers

Qualsys

- **Most helpful Transport Layer Security checker.**
- **Very good at checking cipher suites and versions of TLS**

Postman

- **Most helpful backend function checker pre-client implementation.**

Client side

- Implemented by C# forms on a VS.
- Almost like Java but you don’t have to implement JFrame

The Revised Phase I Plan

- **TLS is configured in AWS for A+ demonstration but the application will be made with the Heroku automatic grade A configuration**
- **Back end messaging functions are made with Ruby implemented on Rails via virtual machine.**
- **Authentication token was implemented in the back end with JWT.**
- **Storing user data through SQLite3**
- **Front end functions (client) are made with C# implemented on Microsoft Visual Studios 2015 via Windows OS.**
- **Planning on PGP encryption/decryption with C# Cryptography library.**
- **Original plan for key exchange: email or physical exchange (USB, or looking) to avoid the server to prevent Man in Middle. In progress.**

Demo