ELEC-H417 - Project Assignment

Denis Verstraeten, Wilson Daubry
{denis.verstraeten, wilson.daubry}@ulb.be

November 24, 2021

1 Introduction

The goal of this project is to design and implement a basic chat app enabling private communication. It will be an opportunity to have a practical understanding of different concepts in cryptography and networking studied in the theoretical classes.

This app should be based on central server allowing an arbitrary number of clients to create an account and have encrypted conversation between one another.

We ask you to be creative as well, and to implement any features that feels funny and/or interesting to you. Feel free to go any directions, as long as you match the requirements.

2 Requirements

2.1 Server

The app will rely on a centralized architecture, meaning that users using their clients will never send packets directly to each other. The central server will have to handle the following tasks.

Client registration, authentication and login A new user should be able to create a new account with a username and a password. A registered user should be able to login to the app using the challenge-response authentication scheme before using it. Think about privacy here, is it a good idea to send and store passwords in plain text?

Start a conversation Create a store a conversation between two users based on their username. For each conversation, a new symmetric key must be created. The key generation and agreement process must be facilitated by the central server by forwarding protocol messages from one party to the other. Despite this and the fact that the server should store the messages, it should never be able to read the plain text messages.

Conversations The server should be able to get and serve messages to the users to ensure the communication.

2.2 Clients

To interact with the app, the users will need to use a client. The client will take care of communicating with the central server and to handle the user's inputs, by being able to perform the following tasks, many of them being symmetric to those of the server.

Registration, authentication and login The client should be able to register using a username and a password. A registered user should be able to login to the app using a challenge-response authentication scheme.

Start a conversation The client should be able to start a conversation with another existing user by addressing them through their username. These two clients should then be able to privately agree on the symmetric key that will be used in this conversation.

Conversations The client should be able to post and fetch new messages from the server in order to have conversations with other users.

2.3 Remarks

The language used to implement the project does not matter, feel free to code in whatever language you are comfortable with.

If something is not specifically mentioned in this assignment, it means that you are free to take any decision concerning that point.

You are allowed to use libraries to do this project. For some parts like cryptography and networking, you even should use them, since it is a good practice. Do not reinvent the wheel. However, it is forbidden to find a off-the-shelf chat library and just instantiate it, or to copy-paste the source code of a chat app.

You can copy-paste code coming from online forums, but if you do so, please specify it clearly in the comments of your code for the sake of intellectual honesty.

Before diving into the coding, take some time to think about the architecture of your project, as well as about a suitable communication protocol. This will save you time later as it will dramatically reduce the time you will need to debug.

You will need to come up with a mechanism to make the IP address of your server available to your clients, you have the choice on how you want to design and implement this, as long as it is explained and documented.

3 Deliverable

You should deliver your code as well as a report. Both need to be submitted via git, as explained in Section 3.3. The deadline for the git submission is the **22$^{\text{nd}}$ of December at noon** as stated in the `ELEC-H417_Labs_Organization.pdf`.

Two files will have to be present in the repository:

- A `README`
- Your report in `pdf` format.

For the rest, the structure is up to you, as long as it is explained.

3.1 Code

Your README file should be clear and understandable for someone who has not taken part into your project[1]. It should explain how to run the code, whether libraries need to be installed and/or imported, if so how to do it. It should also state whether there are known compatibility issues. Finally, it should present the different features of the app and how to use them. All of these guidelines are very common and are good practice when you want to post code online. It should not be a report, but a file oriented towards the actual usage of the app.

Your source code should be well structured, clear and easy to read. You should use comments whenever necessary and keep. You should keep in mind the people reading it were not with you when you wrote it.

3.2 Report

In the report, you will have to discuss the following points:

Architecture Explain the architecture of your code, list the decisions that were taken and give a justification if this is relevant. If you want, you can include diagrams.

Innovation and creativity You are encouraged to show some creativity and to go further than the requirements of this assignment. You should discuss your added features in the report.

Challenges List the difficulties faced during the project and how you solved them.

You can choose the length of the report, in the sense that we want you to discuss anything that feels important to you regarding the project. We would suggest a length between 5 to 10 pages.

3.3 Submission

We want you to use the git versioning system throughout the your development, as well as for the submission of your deliverables. You can use any git host that you want, such as GitHub or GitLab.

To submit your project, you have to send one e-mail per group to the teaching assistants with a link to your repository.

3.4 Grading

As said previously, we want you to show some creativity. Therefore, if you fully match the requirements, you will be graded 16/20. The mark will be computed as follows:

$$\text{Grade} = {}^{3}\!/_{4} \cdot \text{Code} + {}^{1}\!/_{4} \cdot \text{Report}$$

$$\text{Code} = {}^{4}\!/_{5} \cdot \text{Required} + {}^{1}\!/_{5} \cdot \text{Originality}$$

$$\text{Report} = {}^{4}\!/_{5} \cdot \text{Required} + {}^{1}\!/_{5} \cdot \text{Originality}.$$

[1] For example, your teaching assistants.