ASCII Chat

Authors: Lucas Chen (lmc336), Danny Qiu (dq29), Chesley Tan (ct353)

Plan for Meeting

We plan to have at the minimum biweekly meetings in Cascadilla Hall in order to update each other on changes in plans or status of the project, as well as discuss details of subsystem design and implementation.

System Proposal

Vision

For our final project, we will be building a terminal-based peer-to-peer video chat application with end-to-end encryption. It will feature text-based video rendering, audio support, and messaging support. We envision that our application will have diverse applications, such as usage in video conferencing or collaboration in bandwidth-constrained or graphically-constrained environments.

Features

Our program will support/have:

- Colorized text-based video-chatting with up to 4 peers
- End-to-end encryption of video, audio, and messages
- Ephemeral messaging capability - messages are not permanently stored and are encrypted during transmission
- Audio/Voice chat
- Cross-platform compatibility

Description

To video-chat with peers, our program will accept an IP address or URL of the hosting user, who acts as the gateway for other users to join a video conference. Each user within the video conference will act as both a server and a client in that each user's server sends its video to other clients, as well as receives video from other clients. The video will be displayed in ASCII, where the variation of characters will be used to represent differing local features within the image. In addition, the color capabilities of the user's terminal will be utilized. For minimizing bandwidth, colors in the video will be constrained to a set of 256 predefined colors that span a wide range of the color spectrum. Each user's server will locally process the video input from the user's camera to generate an ASCII representation of the image, as well as compress the colors in the image and resize the image appropriately. To support video conferencing with up to 4 peers, the user interface will automatically re-orient itself to maximize screen usage and equalize screen space for each video stream. Users will also be able to send text messages to each other, as well as transfer audio from the user's microphone to other users. Finally, our program respects user privacy as all data that is transmitted will be end-to-end encrypted, and no data will be persistently stored.