Aneesh Agrawal and Paige Studer
6.111 FA2017 Final Project Checklist

Commitments:

RS-232 Serial Communication: Aneesh

This involves a series of modules which together compose an RS-232 input/output stack, allowing us to display prompts to the connected serial terminal, receive characters and perform remote echo for those characters, which will allow the user to enter the first factor: their username and password. The sending and receiving pipelines are completely separate at this point. The sending pipeline is a divider module which generates a baud_enable signal, as well as a sending minor FSM which receives a char (8 bits) and converts it to an RS-232 frame.

The receiving pipeline is more complicated: the incoming signal is synchronized, oversampled (at 8x the baud rate, controlled by another divider module), downsampled (with feedback that ensures we restart a downsampling set of samples when we receive a fresh start bit), and finally an RS-232 frame receiver, which receives a stream of input from the downsampler, checks the validity of the frame and posts a character when it receives a full frame of data.

XVGA Display: Paige

The XVGA display will do multiple items. First it will distinguish between login, authorized, and not authorized states by displaying those states on the screen. It will also display characters that are received via serial, such as the username input characters on the screen. The main module called the address calculator which will determine what ROM to look at as well as the address in that ROM to use for the display. It is a basic display and reflects what is happening in the login.

Main FSM - Paige

The Main FSM module will keep track of the login and will send appropriate information to the XVGA and RS-232. It will store input characters in buffers to be used later to check in a library. It will determine whether a person is authorized or unauthorized by looking into an identity database BROM. Lastly, it will send prompts via serial to let the user know what to input on the computer.

Goals:

Second Factor - Aneesh

The second factor will provide the additional security properties of the 2FA system, and will consist of analog circuitry to recognize the input along with a set of digital modules to synchronize, sample, and decode that input. This will most likely be a stack that recognizes MIT ID card taps via RFID, and performs PSK decoding, RFID decoding, extraction of the useful 32 bits out of the full 224 bit repeating data stream, descrambling of the proprietary FlexSecure protection of the cards, and conversion to BCD to recover the actual second factor, in this case an MIT ID number. An alternative might be inputting the ID number on a TV IR remote, and using a similar stack to read and decode the number, requiring that the full number be input within a certain timeout.

Joke Display - Paige

The joke display will be an addition to the XVGA display. There will be a joke database and when a person is approved, a randomized joke will be displayed on the screen.

Line Editing - Paige

To enable line editing we will allow the person who is inputting his/her username and password. For example, if a person hit “backspace” or an arrow key, he/she would be able to go back and change a previously written character.

Stretch Goals:

Pictures - Paige

One stretch goal of this project is to use an SD card to store images of authorized users, and display the appropriate image along with their joke when a person authorizes herself to the system.

Secure Password Authentication - Aneesh

Another stretch goal of this project to use a more secure method of transmitting the user’s password over serial than simply transmitting it in plain text. This will involve writing a small serial wrapper (likely in Python) for the client side, as well as a set of Verilog modules to complete the handshake on the FPGA. Possible approaches include encryption (likely asymmetric), a challenge response system using hashing, or implementing a more complex system such as Kerberos or SSH.

The raw data received by the FGPA will be displayed on the hex display to show that the raw data is indeed scrambled, but that the FPGA is able to recover the original password.