Programming Assignment 4
CS 370

In this assignment you will be building an application that can work with `Google Authenticator` (GA) [Android] [iOS] [Source]. GA uses a Time-Based One Time Password (TOTP) algorithm to generate One Time Passwords (OTPs) which you can use as a potential 2nd factor authentication method.

The specification for TOTP algorithm is defined in RFC 6238. RFCs are documents from Internet Engineering Task Force (ITEF) which is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet.

Google authenticator supports different kinds of OTP algorithms but in this assignment, you are going to implement 30-Second TOTP codes. If you haven't used it ever before, I encourage you to try Google Authenticator for your Google accounts to make sure you understand how it works from a user's perspective.

As a part of this assignment, you have to build an application that can generate QR codes which are readable by Google Authenticator Android App. Functionality wise, I assume, Android and iOS apps are same and are supposed to provide similar codes. In case you cannot test your program against GA Android app, you can specify that along with your submission. You can use any programming language to complete this assignment but I should warn you that finding libraries to generate barcodes might be easier in some languages as compared to other.

You have to submit one tar compressed file with a 'Makefile' that compiles the program if it needs compiling. Also, please add explicit instructions on how to run your program in the README.txt file. Also, explain your implementation briefly in README.txt file. You are allowed to make any reasonable assumptions about your program's functionality as long as you mention them in your README. Your program should work like this:

- `./yourLastName-MP4 --generate-qr`: This command should generate a jpg picture or svg picture of the QR code that encodes the URI GA expects. URI contains secret keys along with the user id required for the TOTP algorithm. Refer to this page here which provides details about the format of URI and how to add extra information in URI.

- `./yourLastName-MP4 --get-otp` : This command should generate an OTP which would match the OTP generated by the Google Authenticator for that 30 second period and print it to the screen, and then waits for the appropriate time and prints the next OTP and continue to go in this manner printing a new OTP every 30 seconds in sync with Google Authenticator. **Using python or other OTP libraries will NOT get you any points.** You will have to implement your own OTP generation function following the specification from the RFC.