Brian Huang
CS370
Homework 4

For this homework I followed the algorithm described here in RFC4226. This page describes the HOTP algorithm and how it works. I used this as a guide for my code. Here is the step by step of what my code does.

1. First a the shared key is generated. In the program it is hard coded as "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
2. Next the counter is generated. Since this is TOTP, instead of a counter I used a timestamp. For the timestamp I used unix time provided by time.time() in python. This is then divided by 30, then floored. This makes sure that it will stay the same for 30 seconds.
3. A new HMAC-SHA1 string is then generated with the shared key and the timestamp. The program create a 20 byte string.
4. The HMAC-SHA1 string is then dynamically truncated. To produce a 4 byte string.
    a. The dynamic truncation first takes the last nibble of the HMAC-SHA1 string and sets that as the offset.
    b. The offset is then used to generate a four byte string. The four byte string starts at the offset so the string will be: HMAC-SHA1[offset] + HMAC-SHA1[offset + 1]... +HMAC-SHA1[offset+4].
    c. The function then does some number conversion to convert from hex to dec then to bin. The most significant bit is then masked to 0 to avoid any confusion with signed vs unsigned modulo.
    d. This 4 byte string is then returned.
5. The 4-byte string is then converted into decimal, then modulo with 10^6. The value is 10^6 because we want 6 digits for the code.
6. If the result of the modulo operation is less than 5 digits long it is padded with 0's until is is 6 digits long.

My program will generate a six digit key every thirty seconds. However, it does not match with what is produced by google authenticator. For this homework I followed RFC 6238, RFC 4426, and the wikipedia page for google authenticator as a guideline.

Works Cited

“Google Authenticator.” *Wikipedia*, Wikimedia Foundation, 3 Dec. 2017, en.wikipedia.org/wiki/Google_Authenticator.

“HOTP: An HMAC-Based One-Time Password Algorithm.” *IETF Tools*, tools.ietf.org/html/rfc4226.

“TOTP: Time-Based One-Time Password Algorithm.” *IETF Tools*, tools.ietf.org/html/rfc6238.