



Network Security Final Project

Dylan Tivnan Liam Twomey Jonathan Vasallo



Intro

Idea 1: Client to Client Communication private chat room

Idea 2: password safe (Which contains all website private info that you enter in the database)

-Need session key (derive it from both passwords they use to log in)

1. SSL Echo Server model
2. TOTP done
3. End-to-End Confidentiality AES GCM Encryption
4. Session keys used for current users
5. MITM + Replay Attack accounted for

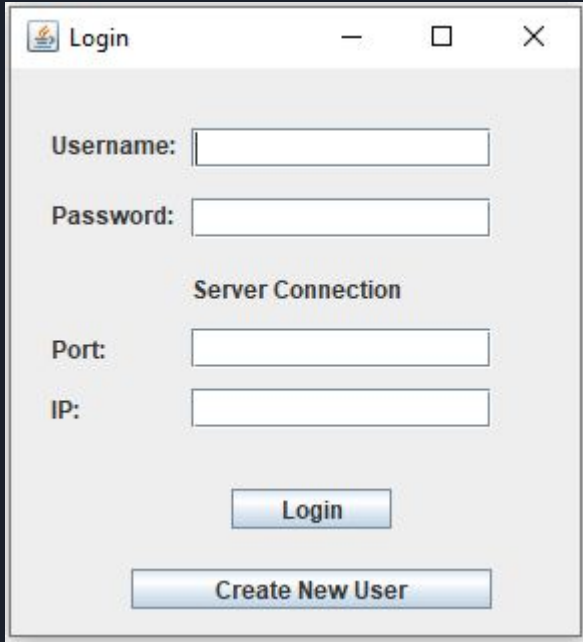


Databases

2 Flat Databases

- passBase
 - Login GUI Database
 - Stores Username
 - Stores Password Encrypted
 - Stores Users IV used with the Key
 - Stores Ket SCRYPT from the Password
 - Stores Users Personal IV
- passWordSafe
 - Password safe GUI Database
 - Stores the Website
 - Stores the Username Encrypted
 - Stores the Password Encrypted

The GUI's - Login GUI

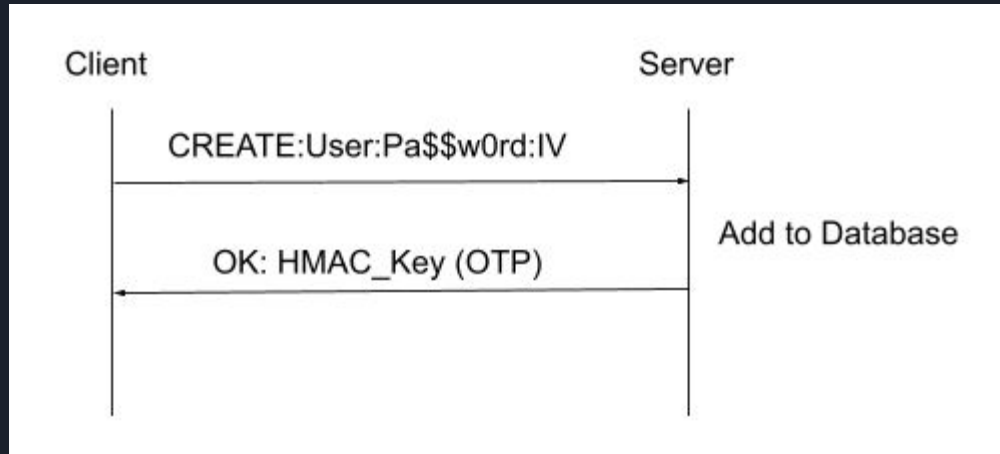


The screenshot shows a window titled "Login" with the following elements:

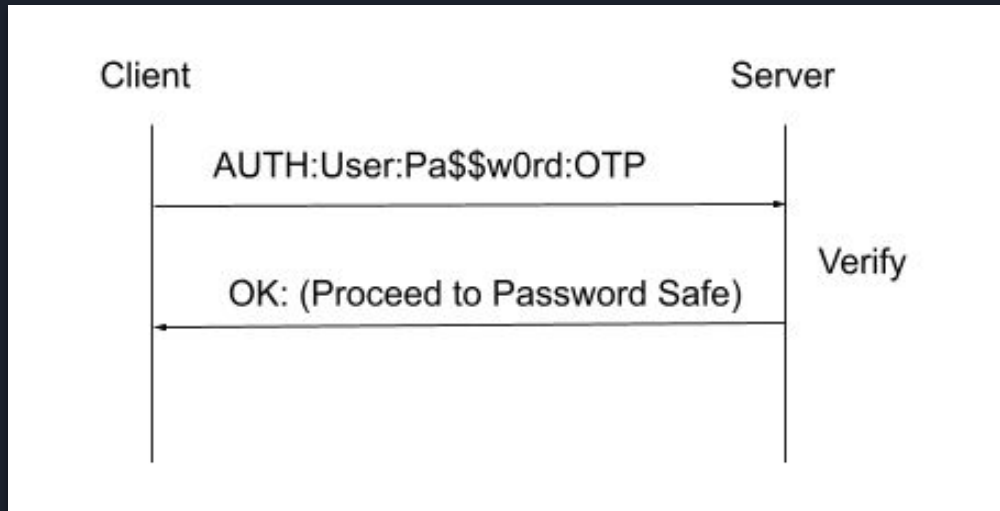
- Username:
- Password:
- Server Connection section:
 - Port:
 - IP:
- Login Button
- Create New User Button

- Port Number and IP are used for Client connection
- All fields must be filled out in order for login
- Login Button
 - Must fill out Port and IP fields for SSL Server
 - Username, Password and IV is sent to the server
- Create New User Button
 - Dialog Boxes ask for Information
 - Username and Password sent to server and is stored in login database
 - OTP HMAC key is sent back to user for Login
 - Must Type in a Port number and IP for Creating a new user
 - Needs to access the server to store new users info into database

CREATE protocol

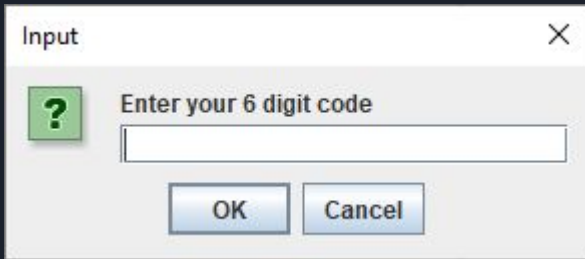


AUTH protocol



The One Time Password

- After creating a new user, the user is given the string shown below
- When pasted into the website given with the settings: SHA1 and timeout, they are given an authenticator that generate one time passwords for them
- After logging in the user is prompted to enter the OTP, and if they enter it correctly they are given access to the password safe

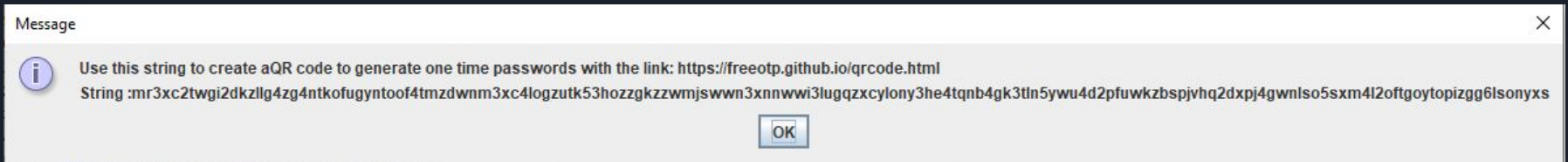


Input

Enter your 6 digit code

OK Cancel

A dialog box titled "Input" with a close button (X) in the top right corner. It contains a green question mark icon on the left, followed by the text "Enter your 6 digit code" above a text input field. Below the input field are two buttons labeled "OK" and "Cancel".



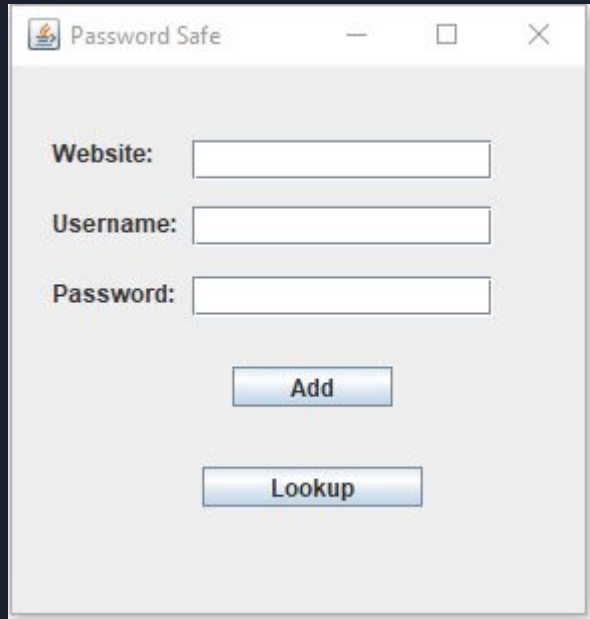
Message

Use this string to create aQR code to generate one time passwords with the link: <https://freeotp.github.io/qrcode.html>
String :mr3xc2twgi2dkzllg4zg4ntkofugyntoof4tmzdwnm3xc4logzutk53hoozzgkzzwmjswwn3xnnwwi3lugqzxcylony3he4tqnb4gk3tln5ywu4d2pfuwkzbspjvhq2dpxj4gwnlso5sxm4l2oftgoytopizgg6lsonyxs

OK

A message dialog box titled "Message" with a close button (X) in the top right corner. It contains an information icon (i) on the left, followed by the text: "Use this string to create aQR code to generate one time passwords with the link: <https://freeotp.github.io/qrcode.html>" and "String :mr3xc2twgi2dkzllg4zg4ntkofugyntoof4tmzdwnm3xc4logzutk53hoozzgkzzwmjswwn3xnnwwi3lugqzxcylony3he4tqnb4gk3tln5ywu4d2pfuwkzbspjvhq2dpxj4gwnlso5sxm4l2oftgoytopizgg6lsonyxs". Below the text is an "OK" button.

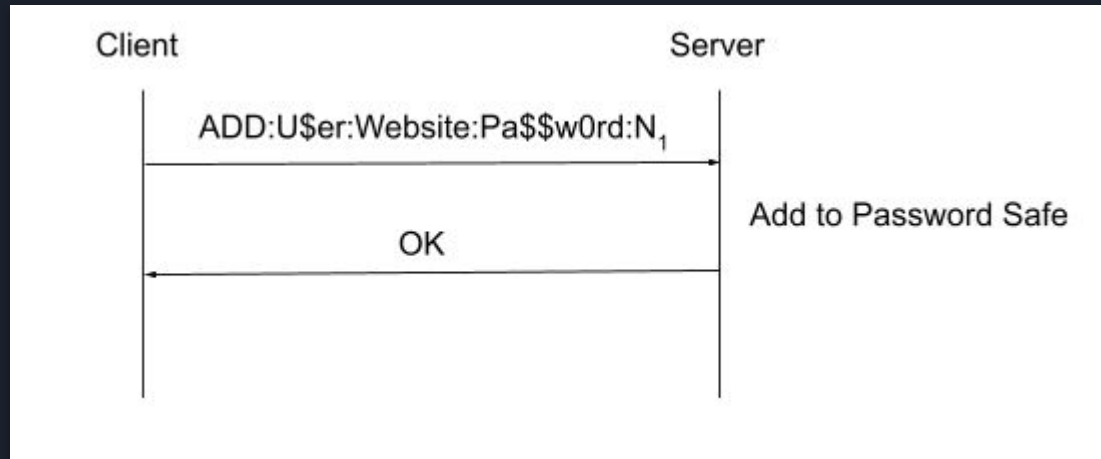
The GUI's - Password Safe GUI



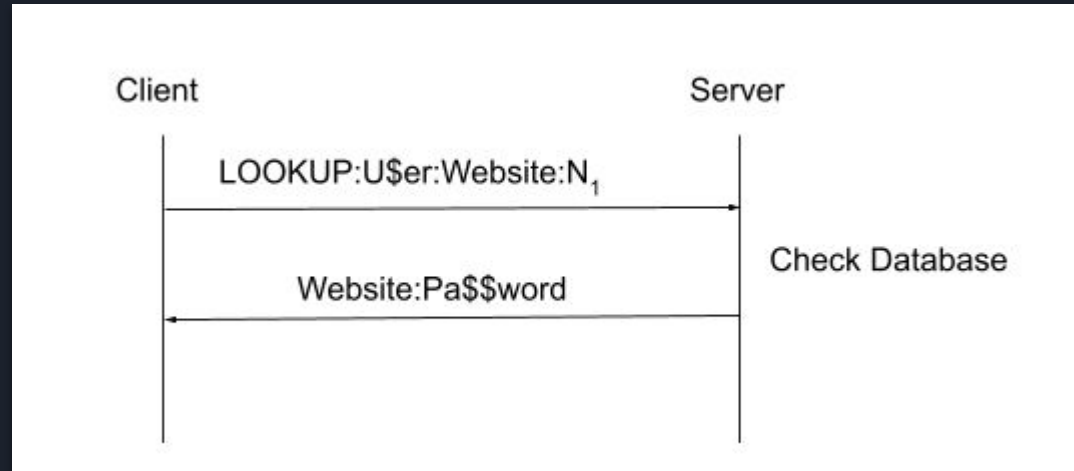
The screenshot shows a window titled "Password Safe" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains three input fields labeled "Website:", "Username:", and "Password:". Below the "Password:" field are two buttons: "Add" and "Lookup".

- When Website, Username, and Password is added, username and password is encrypted and sent to server to be stored in the database
- When Lookup button is pressed, A dialog box will ask for the Website you wish to lookup
- Because you are security logged into the server and password safe GUI, there is no need for username or password confirmation
- Message "LOOKUP" with the website is sent to the database for lookup
- Server sends back the username and password associated with that website

ADD protocol



LOOKUP protocol





Confidentiality and Authenticity

- In order to provide confidentiality, we make sure that all messages that include the username and password, once logged into the actual password safe, are encrypted
- We also have it so that no username or password is ever sent to or retrieved from the server in an unencrypted form, all decryption happens solely on the client side of the program
- Authenticity is provided through the initial login, as well as the one time password
- The initial login provides a baseline authentication method
- The one time password which is user specific provides a secure method of two factor authentication



Security Drawbacks

- If OTP is somehow compromised, adversary has access to users whole password database
 - However, Adversary does not know what websites you have stored into your database
- If adversary can get access to a user password safe, adversary can brute force website look up
- If adversary can figure out cypher method, then adversary can get all information necessary to make session key



Our Challenges

- Including the SSL server to work with our specific GUI was a challenge
 - Server gave us trouble when I would wait for the Client GUI's message
- Having multiple messages sent between the server and client
 - We initially had issues with the client receiving messages from the server, after more than one message was sent by the client
 - To fix this we had to make a method that creates new connection and only sends and receives once per method call
- Early on we struggled to conceptualize what functionalities each class should include
 - For example, should the functionalities of the password safe, adding and looking up, exist within the client or should they be housed elsewhere