|  |  |
| --- | --- |
|  |  |
| Final Project |  |
|  |  |
|  | 05/15/2023CS.4600.02 |
|  | Loc NguyenProfessor Tingting Chen |

1. Keys Set Up

A screen shot of a computer code

Description automatically generated with low confidence

A picture containing text, screenshot, font

Description automatically generated

First, generate corresponding private and public keys for client and server. Then, store these keys in a permission file so we can use these keys for encryption, decryption, signature generation, and signature verification.

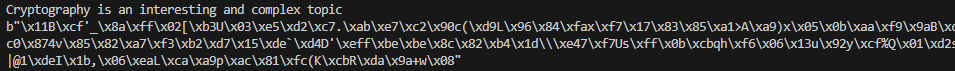
1. Encryption and Preparation

A picture containing text, screenshot, font

Description automatically generated

*Figure 2.1*

First, we get the message to be sent from message.txt. Next, we get the client public key to encrypt the message. Then, we use the public key to encrypt the message that we’re about to send to the client as seen in Figure 2.1.



*Figure 2.2*

In Figure 2.2, we can see the plaintext message and the encrypted message.



A screen shot of a message

Description automatically generated with low confidence

*Figure 2.3*

Finally, we are going to write this encrypted message to a file so the client can access and decode in Figure 2.3

1. Compute HMAC and Send to Client

Now, we are going to compute the HMAC for this message and write it in the encrypted message file as well so the client can authenticate.

A screen shot of a computer program

Description automatically generated with low confidence



*Figure 3.1*

In Figure 3.1, we write a function to generate an HMAC when given a message and a key. The message is read from message.txt and the key is a pre-shared key. This is the assumption that the client and server shared a secret key of “GoneWithTheWind4600!”. Otherwise, HMAC wouldn’t be possible.

A picture containing screenshot, text, font

Description automatically generated

*Figure 3.2*

In Figure 3.2, we append the HMAC to the encrypted messages so client can access.

1. Client Data Retrieval

A picture containing text, screenshot, font, software

Description automatically generated

*Figure 4.1*

In Figure 4.1, the client will retrieve its private key and the encrypted message. We will also initialize the pre-shared key with the server. Next, we read the file encryptedMessage.txt where the first line will be the encrypted message and the second line is the HMAC. 

*Figure 4.2*

In Figure 4.2, this is the data retrieved from the encrypted message file. We can see the encrypted message followed by the HMAC.

1. Decryption and HMAC Verification

A picture containing text, font, screenshot, line

Description automatically generated

*Figure 5.1*

In Figure 5.1, we decrypt the encrypted message and print it out to the console.

A screen shot of a computer program

Description automatically generated with low confidence

*Figure 5.2*

In Figure 5.2, we defined a function that takes in the pre-shared key, the plaintext message, and the HMAC we received from the server.

A screenshot of a computer screen

Description automatically generated with low confidence

*Figure 5.3*

In Figure 5.3, we can see that the original message is received. Then, a comparison HMAC is computed based on the decrypted message and the pre-shared key. Next, we compare that HMAC to the HMAC we received from the server. If it’s the same, we print out a successful message. If not, we print out an unsuccessful message.