Encrypted P2P IM Protocol Document

This program uses AES-256 encryption in CBC mode, with HMAC and SHA-256 for message authentication. The scheme is a MAC-then-encrypt scheme.

1) First, an IV is generated. This will eventually be shared between client and server.

2) Next, two keys are generated: the encryption key and the authenticity key.

Let k_1 = encryption key
Let k_2 = authentication key

3) A server client connection is established using over TCP/IP connection. The server sends the IV to the client so that both client and server know the value of the IV.

4) An HMAC address is computed in a function using k_2.

5) The message is read in from standard input, and concatenated with the HMAC

Let M = message
Let HMAC = Hashed Message Authentication Code

Value stored → M + MAC

6) The program then takes in the concatenated message, the encryption key, and the IV to encrypt the message. Cipher text is generated:

Let k_1 = encryption key
Let C = ciphertext

$C = Ek_1 (M + MAC)$

7) Sender sends the ciphertext, C, to the receiver.

8) The receiver receives the message and computes the HMAC independent of the ciphertext received:

Received from sender → C
$HMAC_{real}$ = some value

9) The receiver then decrypts the ciphertext using the encryption key:

$M + HMAC_{received} = Dk_1 (M + MAC)$

10) Then the receiver obtains the concatenated MAC of the decrypted message (the last 32 bits). Then compares this value with the computed HMAC:

$HMAC_{real}$ vs. $HMAC_{received}$

11) If the values are the same, then the receiver removes the concatenated portion of the message, and displays the unencrypted IM message on the screen.

$M + HMAC_{received} \rightarrow M$

Display M

12) This process repeats on the other instance of the program until the program is terminated.