Tom Ginader

Due: 2/8/17

Project Proposal

MSCS 630

Ensuring Authentication and Integrity using HMAC

For my project, I would like to create an app that sends a message between two parties, and authenticates the message using a keyed-hash message authentication code, or HMAC for short. I've always liked the idea of using an HMAC; I'm not sure why. There's just something really clever and satisfying about its design.

It works by taking a message and a shared key between two clients. This is a key that only those two people know about and own. The message of the sender is then hashed along with the shared key which creates the HMAC. The HMAC is then attached to the message, and then sent to the receiver. When the receiver receives the message, they will hash just the message and not the HMAC along with their copy of the shared key. They will then compare that output to the HMAC sent by the sender. If the output does not match the HMAC, then the receiver will know that the message has been tampered with.

This ultimately aids in two things. It authenticates the sender to the receiver because only those two people should have that shared key. This also provides integrity, the 'I' in the CIA of security. This would also help prevent man in the middle attacks. If an attacker intercepts the message, and tries to alter it, the receiver would know that the original message has changed. Although, this would not prevent someone from reading the message after it has been intercepted. For that to be prevented, the clients would have to use some kind of encrypted tunnel, or they would also have to encrypt their message in addition to sending it with an HMAC. Something like that could be implemented if I end up finishing before the deadline.

The hash used for this project would at the very least use SHA-1 160 bits. But, I will most likely use SHA-256 or SHA-512. If I decide to also encrypt the messages, then I will probably use AES encryption.