The Onions

CSC 426

Project 2: SHA-1

Sean Connor, Chris Harris, Matthew Henschke

Test Vectors:

Christopher Harris, Sean Connor & Matthew Henschke

CSC 426

Project 2

Due Date: 12/06/2018

**Technical Difficulties and Solutions**

During this project we faced a couple of challenges, some were major, others were very minor. The first problem that we faced was starting out. We all spoke with one another regarding how to physically start this project. Being able to conceptualize everything that needed to be done but, not knowing how to write it. To solve this problem, we looked at the Hmac formula as a whole and broke it up initializing everything we needed starting by initializing the output size, block size, inner hash, final hash and both the key pluses. After that we then switched on whatever the mode is and allocated the memory for key+, key+ xor ipad and key+ xor opad. Another big problem we faced was that our output kept on changing for the same msg and key. We spent countless hours debugging trying to figure out why this was happening. Finally, the problem was right in our faces. It was the memcpy function that we were using that was causing the output to be different each time. As a solution to this all we had to do was use strcat instead and it fixed it. The final challenge that we faced was figuring out how to change characters to hexadecimal. To solve this problem we simply casted the value going into printf using "%02X" as a parameter.

Christopher Harris, Sean Connor & Matthew Henschke

CSC 426

Project 2

Due Date: 12/06/2018

**What we have learned from doing this project.**

When we first began this project, we first had to thoroughly understand how SHA1 worked. Using the notes in class and traversing through the program it allowed us to see how SHA1 functioned. After that we then had to fully understand Hmac and how SHA1 is used with it. Having some prior knowledge from the lectures in class we were able to program the Hmacsha1 using a string and by using a file. While programming the Hmacsha1 using files we found an efficient way to read in a lot of different types of files. This led to us learning that we could do this with various types of files.