Jared Faucher & Jonathan Penney

CS51 Final Project: Task 4 - Functionality Checkpoint

TF: Ben Shryock

Progress:

A significant portion of the core functionality has been written for two methods of Shamir's Secret Sharing Scheme. A naive implementation, which is not fully information secure, and a Finite Field Arithmetic implementation, which is fully information secure, has been written, but both require additional testing. Currently, the product will be accessed via the terminal window and will give the user an option to access either the encryption or decryption functions. This allows for a cleaner user experience, rather than having to call separate programs. The naive and Finite Field implementations both require additional testing.

Problems:

Several problems have arisen throughout this project. During the reconstruction phase, given the threshold number of keys, we encountered an issue in dealing with floating value imprecision. In our case, we have an polynomial as an int list of coefficients, which must be divided by an int. We discovered this would produce buggy results in the end where the secret generated may become rounded.

This problem was solved by writing a few additional functions to determine Least Common Denominators, given a list of ints. However, this introduces another error, which is we are reaching the bounding limits of ints.

There is no easy way around this, except to re-implement the types using BigNums instead of ints. Hopefully PSET 3 will come in handy.

In addition to these problems, we've encountered some trouble with the modular arithmetic calculations used in the Finite Field implementation. We have solved these problems by adding additional functions to find the multiplicative modular inverse of a number and to mod the coefficients of a polynomial.

Lastly we have faced a problem getting to our functions to compile from separate files. Because our naive and implementations share very similar interfaces hope to figure out a way to factor out code from our current implementations to make our code more organized and readable.

Teamwork:

Jared had finished the OMG: COWS! assignment early and was able to make significant headway on the core functionality. Together, we have met several times to discuss the methodology of Shamir's Secret Sharing Scheme, and the differences between the

secure/insecure method. With a strong mathematics background, Jared has been instrumental in coming up with key functions to support the calculations, whereas Jon has focused on QA/testing and user-interface experience. Going forward, Jon and Jared plan on integrating BigNums to allow for much larger secrets to be allowed. Additionally, our goal is to expand the interoperability and allow the user to input a string as a secret.

Plan:

Our plan for this weekend involves Jon and Jared testing our current naive and Finite Field extensively to find any bugs that have been missed so far. In addition to this Jon is planning on working out how to factor out code into separate files and compile our two naive implementation and Finite Field implementation simultaneously from the same Makefile and using the same interface signatures. Our goal for next week is to translate our naive and Finite Field integer implementations into implementations using BigNums. Jared and Jon will be working at this task together, mapping out the functions used in the integer version of our code into their BigNum counterparts. Lastly if time is permitted before Thursday night, Jared and Jon will be working together to add a function to convert a string secret into a BigNum by mapping each character in the string to its ASCII-equivalent, and vice-versa for the decryption phase. In addition to this Jon and Jared will be working together on the video requirement for this project to be finished Thursday night.