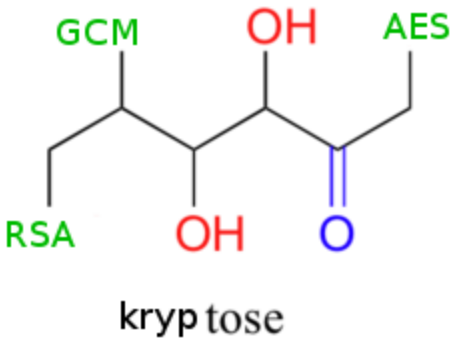
Kryptose™

Sprint Report: Beta



Personnel: Jonathan Shi (js2845), Antonio Marcedone (am2623), Alexander Guziel (asg252), Jeff Tian (yt336)

Activity Breakdown

Jonathan:

I developed the functionality for the user authentication table to save itself to disk, and integrated the user authentication table in with the rest of the server logic. I wrote unit tests for the user authentication table. I overhauled the communication protocol to send checked exceptions to the client. I started writing tests for the server socket handler. I tackled various quick TO-DOs and bugs. I worked on Kryptose™ documentation.

I'll estimate 10-15 hours this sprint. The time was used effectively.

Antonio:

Building on what I did for the first sprint, my main task in this sprint was to choose appropriate cryptographic primitives for the various tasks, and to make an interface for the java cryptographic libraries that the rest of the team could use transparently.

In particular, I focused on authentication, and designed (after a lot of research) how to derive the two cryptographic keys (one for authentication with the server, one for encryption of the list of credentials) from a single master password the user enters. On the server side, I designed the data structures to store the (properly hashed) master passwords, and the functions to authenticate the user.

I also started designing JUnit tests for the parts of code that I wrote during the first two sprints, mostly adapting the manual individual tests that I did while designing the code. I finally helped testing the client and the server and suggesting improvements and bugs to fix to the others.

Due to the high workload I had in this part of the semester, we decided to postpone the optional features to the final, and I actually estimate around 20/25 hours of work. My efficiency in writing code and using the cryptographic libraries also increased considerably over the previous sprint.

Alexander:

I implemented the server side component for retrieving user logs. I also made a Formatter which writes the logs in tamper-proof format and changed the logging on the server to use this Formatter to write the logs. I wrote tests to assure this works correctly. I also wrote a utility to create a secret key for tamper-proof logging from a password that the admin supplies. This utility is also able to read logs when given the admin password. This was necessary to do but took longer than expected so I ended up having to postpone other features to the next milestone. My estimated hours spent is about 20.

Jeff:

I further developed and made updates to the client side command line user interface and the backing functions used by the interface that allows the client side to function. This includes writing the client side logic for account creation and logging in/out, using the new master password provided in order to link up with the new cryptography interface which now includes the authentication-related functions. To accommodate the new features added in the beta, my role also included making new request types for the new interactions with the server like requesting account creation and user logs. I also revamped some existing commands to make them more user-friendly and flexible, such as changing the way passwords are prompted to allow them to contain special characters like spaces. To test the client side CLI I manually tested all of the commands and major command sequences.

Productivity Analysis

The Kryptose™ team productivity for this sprint is of serious concern. Our employees fail to be dedicated to the project, and flippantly disregard their expected work hours. If this absenteeism continues, they will have to be fired, and new talent acquired.

The project personnel would improve their productivity and work standards by learning to use `git branch` and pushing only tested and functional code to the respository's main branch… well except for Antonio who does a good job of this already. Employee-of-the-month for him!

The team has succeeded in implementing the required minimum of authentication and secure server logging, as well as a few more user story scenarios and some unit testing. It is unclear precisely what the team had intended for this sprint, but it was likely never more than one or two user stories more than this.

Writing unit tests took the team less time than expected. Fixing places where exceptions were not handled correctly took more time than expected. More code review could help problems not get out of hand.