**Computer Organization**

**Project**

**Written by:**

**Dalton Sumrall**

**Ryan Thimons**

**Jacob Lahav**

**Eduardo Mata**

**Project Flow**

First step was to set up the stack, then make room for any variables that we used in the project.

Second step was to construct and debug the loops, starting with the rounds loop, followed by the indexes and hop counts. Afterwards we constructed the XLOOP, then we worked on getting the appropriate values from the file and the key file to XOR them together using the index and hop counts. Lastly we created the bit manipulations.

**Functions**

**ROUNDLOOP: Cycles through the rounds.**

**HOP1NOTZERO: Acts as an if statement.**

**HOP2NOTZERO: Acts as an if statement.**

**XLOOP: Cycles through the inner for loop.**

**INDX1OKAY: Acts as an if statement**

**REVLOOP: Used for mirroring the byte.**

**INDX2OKAY: Acts as an if statement.**

**Project Complications**

Initially we analyzed the C functions and went over together how we would create the functions and how we would perform the bit manipulations. Our hardest bit manipulation was swapping the half nibbles. After we got encryption and decryption written, we found that decryption would only work on odd amounts of rounds (i.e. 1, 3, 5, 7…) and would not decrypt correctly with even amounts of rounds. We found that our index and hop count operations were incorrect in the decryption method and we found that we needed to decrement the current round when calculating the index and hop count. Also getting visual studio to work was a pain.

**Contributions**

We all met every Tuesday/Thursday at 3 PM to work on the project. We used slack to communicate with each other if we got stuck or to keep each other updated. Over all we worked together to break down the program into small manageable pieces and solved them one at a time. When we started the decryption portion we figured out what C code we needed to shift around first and then moved the appropriate assembly code around.

Dalton:

Team captain, built the stack, pieced all the assembly code together and commented.

Eduardo:

Contributed ideas on the implementation of the stack.

Jacob:

Researched instruction sets for bit manipulation.

Ryan:

Flipping encryption to decryption, advising on assembly code.

**Conclusion**

The project proved to have its challenges, but we feel that it has effectively taught us to use

in-line assembly, as well as use the manual when we needed to research the instructions syntax and functionality.