MSCS 630L 711 Security Algorithms & Protocols
Professor Kippins
Assignment: Milestone
Due: April 17, 2022
Kerry Lyon
Github: klyon0517

Abstract:
(A brief overview of the paper.)

This project will combine the AES encryption process with animation in the Babylon.js web rendering engine. Tentatively titled “BabylonAES”, the goal is to provide a real time 3D visualization of how encryption works from plaintext and key to ciphertext. The idea is to have various planes with text moving around the screen and changing their letters and hex values depending on which stage they are at in the process. The end user will be able to rotate the camera, zoom, and pan to view the display from any angle.

Introduction:
(Describes the motivation of this work and outlines the rest of the paper.)

Having a great interest in game development, particularly in the tools and software, I wanted to see if cryptography could be combined with a real time rendering engine. As a full stack developer, I wanted to work with JavaScript and a web based engine to illustrate AES. I chose Babylon.js for this task. The goal is to create an animation of the process that is easy to follow and entertaining.

Related Work:
(Describes what other researchers in the same area have done,
and how they perhaps could be improved.)

There is an excellent AES Rijndael animation that breaks down the process in an easy to understand manner (Zabala, Enrique. 2017). This is a linear 2D animation. My work is a real time 3D animation, so it differs quite a bit in that the whole process is occurring as the viewer explores the scene.

Methodology:
(Describes what is the approach taken in this paper.)

An iterative approach has been taken in the conversion of various labs from Java to JavaScript. Once the results are returned as expected, layers of animation are applied to each step. The final goal is the conversion of lab 5 (AES encryption). And the final animation is built around this process.

Experiments:
(Describes the experiments performed, including details on the data used.)

The experimentation began while converting Java to JavaScript. Fortunately, this wasn’t too difficult. Starting with lab 1 and lab 3 as practice helped in preparing for the lab 5 conversion. Knowing the end result from the previous labs was a great assistance in reaching the final form.

At the same time, practice with simple animation while learning Babylon.js was ongoing. Features included rendering the text on the plane, adding dynamic textures, world positioning, camera positioning, and animation timing. The creative ideas developed as progress was made on the experiments.

Analysis:
(Examines the results of the experiments and draws some conclusions

about their significance.)

So far the conversion of lab 5 to JavaScript has been completed. It took some trial and error to reach this goal. Fortunately, comparing the output for various functions from the Java version to the JavaScript version helped determine where mistakes existed.

Babylon.js animation basics were learned via tests on lab 1 and lab 3. Now they are being applied to this project. My favorite so far is witnessing the calls to SBox and filling in which sections are used.

I'm noticing slow down due to how many planes are being rendered and the computations being run. No time for optimization yet.

Conclusion:
(Summarizes the paper and its findings.)

References:

Zabala, Enrique. (2017, November 27). AES Rijndael Cipher explained as a Flash animation. https://www.youtube.com/watch?v=gP4PqVGudtg

Zabala, Enrique. (2017, November 27). Rijndael Cipher (winner of the AES selection). https://formaestudio.com/rijndaelinspector/archivos/Rijndael_Animation_v4_eng-html5.html