Tony Alexander

Advisor: Randy Appleton

Committee: John Sarkela & Hadi Shafei

Analyzing the Randomness of the Keccak 512 Hashing Algorithm using a Provably Fair Gambling Site.

I have always been fascinated with random number generators and how a computer is able to *replicate* randomness. Another topic that has peaked my interest is the math behind how online casinos generate their randomness in their virtual games. After looking deeper into both of these topics, I found out that many of these casinos use hashing algorithms to determine a random number that is plugged into their own equation to determine an outcome. For this project, I will be looking at how wtfskins.com casino site uses the Keccak512 hashing algorithm to create randomness and perhaps in the process will find data that supports a not-so-random occurrence of numbers that could be used as an advantage against the house.

My main approach for looking deeper into this hashing algorithm is to generate millions of hashes that are then fed into wtfskin's formula for determining a multiplier on their crash game (Testing data after the hash is fed through the crash formula does not mean I am testing the website's formula, because everything is still based off the hash). Once enough numbers have been generated, I can analyze their data by running it through different combinations of multipliers along with using a martingale betting strategy. Because of such a large data set, this will be repeated on different spans of the crash numbers to determine if there is consistency between wins and losses. Additionally, if a pattern does show where this is a possibility to make money, a script will web scrape the website to automatically place bets that follow the strategy described.

The language that I will be using for this project is Python. I am familiar with Python but not the process of web scraping. In order to scrape data off the website as well as automate the betting process, Python offers the simplest approach to writing code quickly and easily. Upon first running the script that analyzes the data collected, I noticed that running through a file of 1 million numbers 500,000 times obviously took an extremely long time. So, I would like to start learning how to utilize multiprocessing as well as taking advantage of GPU cores when dealing with big data. Once all of the data has been processed, another good learning experience is to analyze the data visually as in a 3D array. This would allow me to see if there are repeating patterns and occurrences between the datasets that I am testing. The frameworks and software that I want to utilize to generate this 3d representation of the array would be matplotlib, as well as a graphics simulator to be able to make changes on the fly to the data for better testing, viewing, and analysis. Because I am not sure if there is a Python framework to generate this 3d visualization that I am looking for, this may be something that I will have to code up myself for custom options.

Overall, the gist of this project is to learn how to analyze and gather data, run tests on said data, visualize your findings, and utilize those findings to maybe be used in a practical scenario. **My goals in this project are to learn web scraping, utilize GPU cores in Python, learn how to analyze big data visually, create 3d graphics, and prove/disprove the Keccak512 hashing algorithm.** Keccak512 is known by many to be secure but the steps provided above will be able to provide more insight and knowledge to me as well as others on if it actually produces *completely random* results. Doing so utilizing the casino website mentioned above will (in my opinion) make this project much more fun and exciting since there could be a benefit in the end!