Cullen Watson

CS2353 Fall 2021 Project (Logic)

My program demonstrates a basic integrity check function. The user inputs a string, and then the program will output whether the string has an odd or even number of 1 bits. The way I implemented this was by chaining together an exclusive or (XOR) on every underlying bit in the string. For a XOR on two bits, it is true only if one bit is a 1 and the other bit is a 0. If you chain an XOR on all the bits, if there is an odd number of 1 bits, the XOR will return true and vice versa. Overall, this integrity check function is a boolean function: $F(X) = X_0 X_1 X_3..X_n$. Boolean functions receive a chain of bits and output 1 resulting bit. The domain of boolean functions is a string of $\{0,1\}^n$ bits and the codomain is {0,1}. Boolean functions are onto because each element in the codomain has a corresponding X such that F(X)=y, where y is 0 or 1.

The reason I chose this integrity check function is because of its vast use in security and web design. Integrity checks are simple examples of a type of checksum. A checksum compares the underlying bits of two files. This is useful when downloading files off the internet. Most secure websites have checksums on their sites, so you can verify that the file that you download is the file that the website intended for you to receive. In the case of my program, a website could state on their site that the file that on their site has an odd number of 1 bits. Then, once you download the file, you would run the file (in this case, a string) through the program, and if you get a different result for odd/even number of 1 bits, then you know your file has been corrupted in transmission, and therefore, you should not use the application. Of course, there are much more sophisticated checksums out there, but this is a basic one to show the use

of logic in a real-world scenario. Most websites use the SHA256 or MD5 algorithms for checksums.