Lakers Encryption Program

This program was created based off my favorite basketball team which is the Los Angeles Lakers. In this encryption and decryption program it includes jersey numbers and inspiration quotes which is centered around the Los Angeles Lakers. The program is written in python and ultimately includes many techniques to show aspects of cybersecurity. It might not be the strongest encryption program, but it shows the ideas of protecting the data.

Explanation of the Algorithm

Encryption is composed of many steps which are transformation of some sort.

Key Generation with LA_NUM ()

Creates a fixed list of numbers which originate from the jersey number of Kobe Bryant and LeBron James. List serves as the key for shifting and chunking function: [8, 2, 4, 2, 3]

Checksum Calculation with buzzerBeatercheck(txt)

Works by a two-digit checksum that encrypts the text by using a Unicode value which takes modulo 24. Then, the remainder is added and create a two-digit string. Also, it is important to note that the modulo 24 reduces is applied continuously. Checksum is used to make the encrypted data which is later used to verify the decrypted text.

Character Shifting with dribbling (txt, jerseyNum, enc=True)

Function does a basic shift cipher that is applied to each character input. It works by cycling through the text using the jersey number list so when encrypting it adds the number but when decrypting it subtracts it. This is the first layer of making of encryption.

Substitution Cipher with passing (txt, enc=True)

Cipher substitution can be broken down into two parts, a defined map which picks consonants from two sets and a vowel shifting over a position. This being a step that scrambles the data even more after it already being shifted.

Chunk Reversal with fastbreak (txt, jerseyNum, enc=True)

Chunk Reversal allows for the division of text which is divided by the specific jersey number. It works by the text being broken into parts which is determined by the successive number that comes from the key. The chunks are reversed and ordered backward; this shuffles the order of the characters in a segment that adds to the complexity of the algorithm.

Complete Encryption Process in lakerEnc (input, output)

The encryption process works by reading the input file in which it will apply the three functions which are dribbling, passing, and fastbreak. Then creating a checksum number from the final cipher text. This will output a file that will have the encrypted text and have a header that will include a random Laker quote that I have provided.

Decryption Process in laker_Dec (input, output)

In the decryption process the file that has been encrypted will now be an input file as it serves the purpose of going back to how it was originally. It will separate the head and the encrypted data which will then proceed to validate the integrity of the encrypted data by comparing the checksum value with the new one created. It will reverse each of the steps by using fastbreak, passing, and dribbling in the decryption mode. It will then output the file in the designated file.