

Design Document

Functions:

-createTables(specialChars) Argument Types: (list)

This function creates two python dictionaries used to store character mappings representing the values obtained from de-referencing the table 'tabula recta', used for encrypting and decrypting. Two for loops are used to iterate the alphabet (list of characters) twice, using two counters to track row and column numbers respectively. The sum of the counters modulo the length of the alphabet is used to access the character of the shifting alphabet each row, while concatenation of relevant characters is set as dictionary keys. The default alphabet is all printable ascii characters, but with extensibility in mind, the alphabet is extended by the list argument passed. Returns an array of 2 python dictionaries.

-encrypt(key,message,table) Argument Types: (string, string, Python Dictionary)

Vigenere encryption. Message and key strings are converted to lists of characters. The characters in the message list are iterated and an incrementing counter is used to access the key character which is concatenated with the current character in message. This concatenation is used as the key to get the corresponding encrypted character from table which is concatenated to the ciphertext string. Ciphertext string is returned.

-decrypt(key,ciphertext,table) Argument Types: (string, string, Python Dictionary)

Vigenere decryption. Exact procedure used in **encrypt()**, however the concatenation of the 'key' character and the character in the ciphertext happens in the opposite order as in **encrypt()** to get plaintext characters from table. Plaintext string is returned.

-getSpecialChars(input) Argument Types:(string)

Function that creates a string by concatenating each special character (e.g Unicode character) passed in the 'input' string that is not already in the alphabet or already in the string being created. Resulting string will have each unique character in 'input' that is not in the alphabet, this is returned.

-prototype() Argument Types :()

Uses all listed functions to provide simple user command line interface that asks for special characters, message and a key. The message is encrypted and resulting ciphertext printed. Then ciphertext is decrypted and resulting plaintext printed.

Extended Features:

Extendable alphabet (encryption more secure). Also object serialisation so that any python object can be accepted and encrypted, not just strings.