## Encryption/Decryption Program

# Introduction

For the final project, I choose to write an encryption/decryption program. My program: creates a textual user interface that guides the user through its options, loads and saves text files indicated by the user, and encrypts text or decrypts text encrypted by the program. To run it, just type EScrypt into the matlab console. My goal was to encrypt the user’s text such that it is not too easy to crack. The program consists of six functions and one script. The functions are: decrypter, encrypter, getdatafromfile, savefile, pswrdchk, and pswrdget; and the script is EScrypt. Each of these and their operations will be covered in detail.

**EScrypt**

EScrypt handles the majority of the user interface operation via switch/case loops. Within the script, there are several sections: the main prompt the case for the encryption option, the case for the decryption option and the case for “done”, which stops the program. The main prompt requests the users input, which is used to direct them to their desired option (opt variable). The options are handled by the switch/case loop where the case is dependent on the opt variable.

The encrypt case takes the user text and encrypts it. This case has options mediated by a switch/case loop. These options include the input options as well as the options to go back to main prompt and to stop the program. The encrypt case calls on the functions “encrypter”, “savefile”, “getdatafromfile”, and “pswrdget”. It also asks the user to provide text; this text can come from a text file or can be typed in directly. If the user chooses to use a file, the “getdatafromfile” function loads the text indicated file. Once text is inputted the case calls the “encrypter” function. The user is asked for a password, which goes through the “pswrdget” function to establish a connection between the password and the encrypted text, so that it can be used to decrypt it later. Then encrypted text is then saved to a text file of the users choosing with “savefile”

The decrypt case takes the users text and decrypts it, unless the text was not encrypted by this program, in which case it will not be able to decrypt it. The decrypt case can only decrypt text files; therefore, it only has the option to decrypt, go back to the main prompt or stop the program. It calls on “decrypter”, “savefile”, “getdatafromfile”, and “pswrdchk”. The user is asked for a text file. The “getdatafromfile” function scans this file, and gathers the data it needs. Next, the user is asked for the password, which is put through “pswrdchk” to generate the key. With the key and the encrypted text, the “decrypter” function can generate the original text. Finally, “savefile” guides the user through the process of saving this text to file, should they choose to do so.

**Encrypter**

The “encrypter” changes the text into random characters and creates a key that can restore these random characters to the original text. It does this by adding a random number to text’s double vector equivalent, finding a nearby number with a perfect square, and finding the difference between the actual square root and the perfect square root. This difference is the remainder. The remainder and the root become the key and the encrypted text respectively. In addition to the remainder values, the randomly generated numbers are also stored within the key.

**Pswrdget**

Because the key is such a long string of characters, the program then asks the user to enter a password, which is processed through “pswrdget”. The “pswrdget” function generates a vector of numbers called the chk variable. The chk variable is stored with the encrypted text in a file and is used to generate the key. “pswrdget” creates the chk variable by subtracting each character of the password from the characters of the key. It does this with a for loop. Because the key is likely much longer than the password, the password has a different iterater, which resets its value to one whenever it equals the length of the password. The purpose of this function is to shorten the key to something the user can remember.

**Savefile**

This function is used to write text to file. It has two inputs. The data provided in these inputs are separated by a tab when written to the text file. It gives the user the ability to choose where to save the file and what to call it. The user can enter their own filename and location, or they can choose the default values. In order to work correctly “savefile” must test to make sure that the directory provided by the user exists. It does this with an if loop and the exsist function. The In addition, if the user chooses to save the file with the default filename, the program will check to see if this file already exists. If it does, it will add a number to the end of the default file name and save it. “savefile” uses fopen, fprintf and fclose to open and write to file. fopen allows matlab to access, edit, or create files depending on the arguments used. Fprintf can be used to write data to file if a valid file ID is indicated in the arguments. Fclose is used simply to close files matlab is using.

.

**Getdatafromfile**

“Getdatafromfile” takes the file indicated by the user, scans it, and separates strings from integers. The function asks the user for a file, and checks that the file exsists in the same way that “savefile” does. If the file exists, it uses textscan to read the data. Textscan’s arguments indicate what type of data to read, and what type of data to skip. There is a textscan that reads the integers and a textscan that reads the string. These data are stored into separate variables and outputted.

**Pswrdchk**

The integers gathered by “getdatafromfile” make up a vector that is the variable chk . Chk and the password go through “pswrdchk” and generate the key. Pswrdchk is the reverse process of “pswrdget”. It adds the password to chk.

**Decrypter**

The “decyrpter” takes encrypted text and the key and generates the original text. To do this, first the “decrypted” has to separate the random numbers and the remainders that make up the key. Once this is done, it performs the encryption calculations in reverse to return to generate the original text.

Works Cited

"MATLAB Apps." *MathWorks*. MathWorks, Inc., n.d. Web. 02 May 2013. <http://www.mathworks.com/help/matlab/>.