Table of Contents

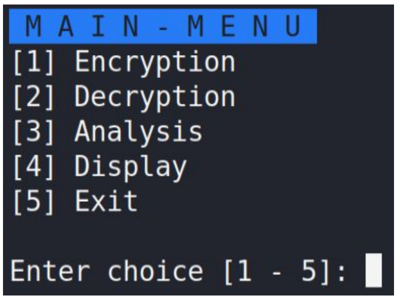
Functions

Overview

Bash version: 5.0.16(1)-release
Environment: Kali Linux

Structure of directory:
The main directory of the program has one script file - ‘main.sh’ - to start the program. All the other functions are in the folder ‘src’. Another folder - ‘test_files’ - is where the input and output files are read from and produced to.

After the ‘main.sh’ script is run, the ‘MAIN-MENU’ is displayed as shown below.



```
 M A I N - M E N U
[1] Encryption
[2] Decryption
[3] Analysis
[4] Display
[5] Exit

Enter choice [1 - 5]:
```

From here the user can enter a function of choice.

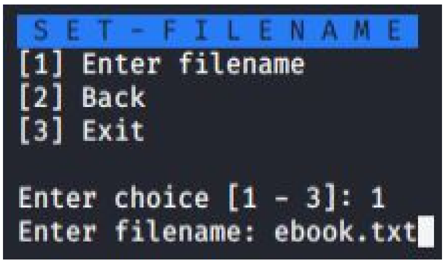
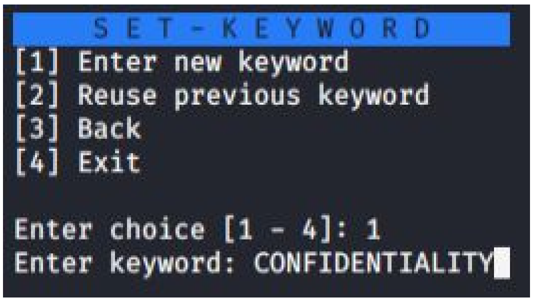
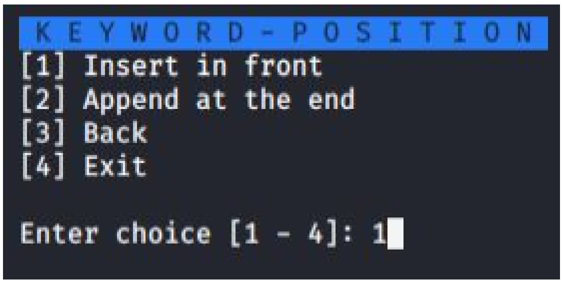
Encryption and Decryption

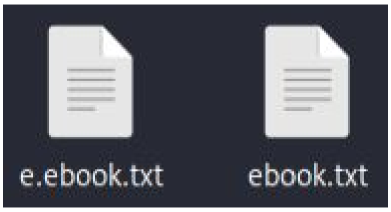
Manual

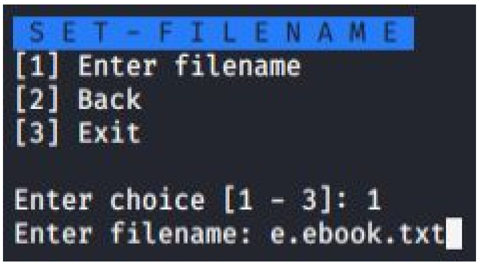
Encrypting a plaintext and decrypting a ciphertext runs on the same set of codes - functions and variables. The only different is the variable $TYPE - ‘E’ for encryption and ‘D’ for decryption.

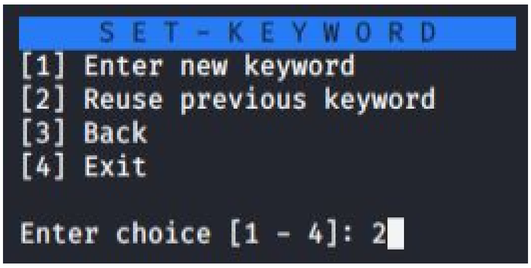
The following manual shows the steps to encrypt and decrypt. The file - ‘ebook.txt’ - is used in the first round of encryption using the keyword - ‘CONFIDENTIALITY’ - at position in front. The result of the first round of encryption is the encrypted file - ‘e.ebook.txt’. The file - ‘e.ebook.txt’ - is used in the second round of encryption using the same keyword at the position at the back. The result of the second round of encryption is the encrypted file - ‘e.e.ebook.txt’.

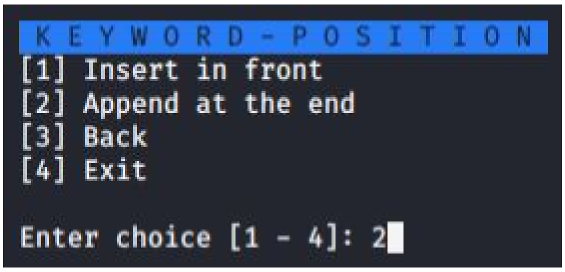
The file - ‘e.e.ebook.txt’ - is used for decryption using the same keyword at the position at the back. The result of the decryption is the file - ‘d.e.e.ebook.txt’.

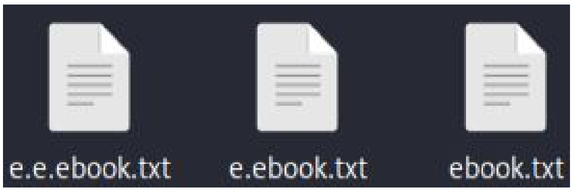
Step	Interface	Description
1	 SET-FILENAME [1] Enter filename [2] Back [3] Exit Enter choice [1 - 3]: 1 Enter filename: ebook.txt	After selecting [1] for ‘Encryption’ in the ‘MAIN-MENU’, the next page is the ‘SET-FILENAME’ menu. Choice of ‘1’ is entered to enter a filename and ‘ebook.txt’ as filename.
2	 SET-KEYWORD [1] Enter new keyword [2] Reuse previous keyword [3] Back [4] Exit Enter choice [1 - 4]: 1 Enter keyword: CONFIDENTIALITY	After entering the filename, if the filename exists, the next page is the ‘SET-KEYWORD’ menu. Choice of ‘1’ is entered to enter a new keyword and ‘CONFIDENTIALITY’ as keyword.
3	 KEYWORD-POSITION [1] Insert in front [2] Append at the end [3] Back [4] Exit Enter choice [1 - 4]: 1	After setting the keyword, the next page is the ‘KEYWORD-POSITION’ menu. Choice of ‘1’ is entered to insert the keyword in front. After the encryption, the next page is back to the ‘MAIN-MENU’.

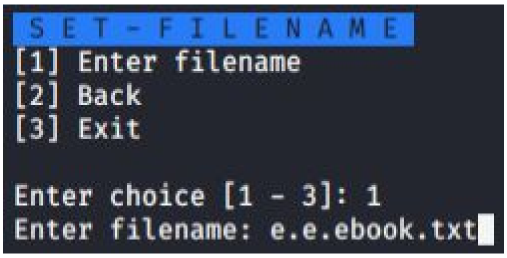
4		Now, there is a new text file ‘e.ebook.txt’ in the test_files folder. The prefix of ‘e.’ indicates that it is an encryption of the proceeding filename.

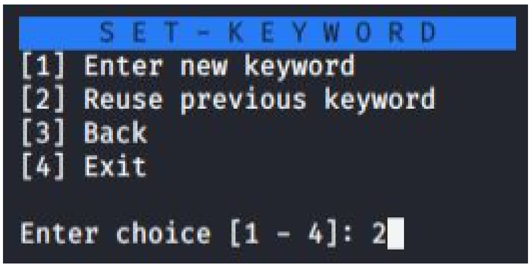
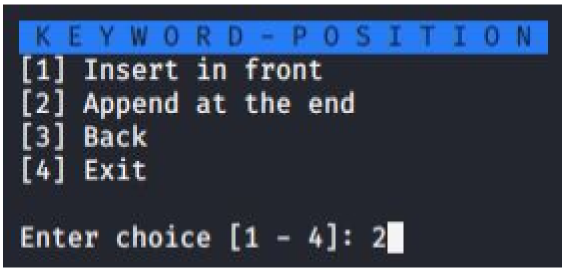
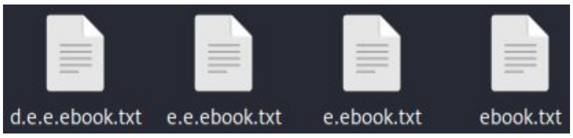
5	`S E T - F I L E N A M E`
	`[1] Enter filename`
	`[2] Back`
	`[3] Exit`
	
	`Enter choice [1 - 3]: 1`
	`Enter filename: e.ebook.txt`
		Navigating back to the ‘SET-FILENAME’ menu after selecting [1] for ‘Encryption’, choice of ‘1’ is entered to enter a filename and ‘e.ebook.txt’ as filename.

6	`S E T - K E Y W O R D`
	`[1] Enter new keyword`
	`[2] Reuse previous keyword`
	`[3] Back`
	`[4] Exit`
	
	`Enter choice [1 - 4]: 2`
		After entering the filename, if the filename exists, the next page is the ‘SET-KEYWORD’ menu. Choice of ‘2’ is entered to reuse the keyword - ‘CONFIDENTIALITY’.

7	`K E Y W O R D - P O S I T I O N`
	`[1] Insert in front`
	`[2] Append at the end`
	`[3] Back`
	`[4] Exit`
	
	`Enter choice [1 - 4]: 2`
		After setting to reuse the previous keyword, the next page is the ‘KEYWORD-POSITION’ menu. Choice of ‘2’ is entered to append the keyword at the end.
		After the encryption, the next page is back to the ‘MAIN-MENU’.

8		Now, there is a new text file ‘e.e.ebook.txt’ in the test_files folder. The prefix of ‘e.’ indicates that it is an encryption of the proceeding filename.

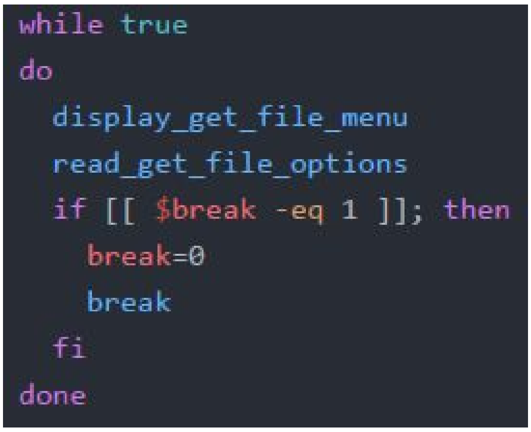
9	`S E T - F I L E N A M E`
	`[1] Enter filename`
	`[2] Back`
	`[3] Exit`
	
	`Enter choice [1 - 3]: 1`
	`Enter filename: e.e.ebook.txt`
		Navigating back to the ‘SET-FILENAME’ menu after selecting [2] for ‘Decryption’, choice of ‘1’ is entered to enter a filename and ‘e.e.ebook.txt’ as filename.

10	 S E T - K E Y W O R D [1] Enter new keyword [2] Reuse previous keyword [3] Back [4] Exit Enter choice [1 - 4]: 2	After entering the filename, if the filename exists, the next page is the ‘SET-KEYWORD’ menu. Choice of ‘2’ is entered to reuse the keyword - ‘CONFIDENTIALITY’.
11	 K E Y W O R D - P O S I T I O N [1] Insert in front [2] Append at the end [3] Back [4] Exit Enter choice [1 - 4]: 2	After setting to reuse the previous keyword, the next page is the ‘KEY-WORD POSITION’ menu. Choice of ‘2’ is entered to append the keyword at the end. After the decryption, the next page is back to the ‘MAIN-MENU’.
12	 d.e.e.ebook.txt e.e.ebook.txt e.ebook.txt ebook.txt	Now, there is a new text file ‘d.e.e.ebook.txt’ in the test_files folder. The prefix of ‘d.’ indicates that it is a decryption of the proceeding filename.

Implementation

After selecting either [1] or [2] from the ‘MAIN-MENU’ as type (Encryption or Decryption), the function for encryption and decryption starts at the ‘set_filename.sh’ script to set the filename. After the filename is obtained from the folder test_files, the script ‘set_keyword.sh’ runs to obtain or reuse the keyword and set position of the keyword either in front or at the back. After type, filename, keyword, and position is obtained, they are passed as parameters to the ‘shift_cipher.sh’ script from the ‘main.sh’ script. The following are the script filenames, their respective important code snippets with a brief explanation.

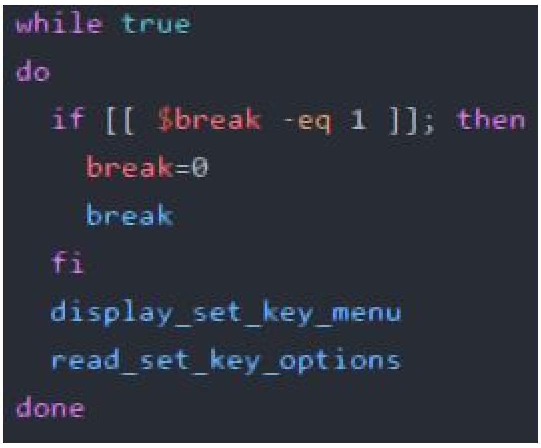
set_filename.sh



```
while true
do
  display_get_file_menu
  read_get_file_options
  if [[ $break -eq 1 ]]; then
    break=0
    break
  fi
done
```

This script runs a loop of displaying the ‘SET-FILENAME’ menu to get the filename.

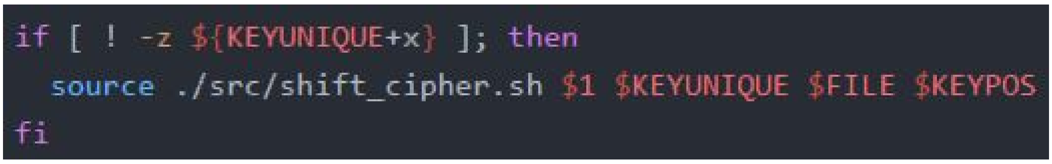
set_keyword.sh



```
while true
do
  if [[ $break -eq 1 ]]; then
    break=0
    break
  fi
  display_set_key_menu
  read_set_key_options
done
```

This script runs a loop of displaying the ‘SET-KEYWORD’ and ‘KEYWORD-POSITION’ menus to get the keyword and intended position of the keyword.

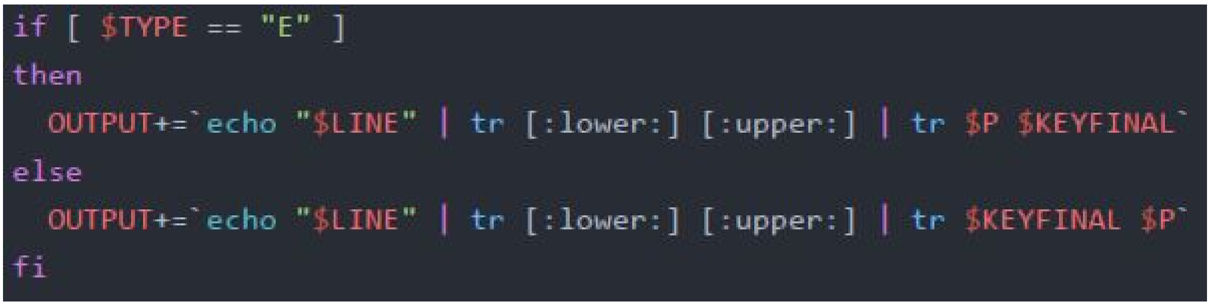
main.sh



```
if [ ! -z ${KEYUNIQUE+x} ]; then
  source ./src/shift_cipher.sh $1 $KEYUNIQUE $FILE $KEYPOS
fi
```

After getting the type, keyword, filename and keyword position, it is then passed to the ‘shift_cipher.sh’ script from the ‘main.sh’ script.

shift_cipher.sh



```
if [ $TYPE == "E" ]
then
  OUTPUT+=`echo "$LINE" | tr [:lower:] [:upper:] | tr $P $KEYFINAL`
else
  OUTPUT+=`echo "$LINE" | tr [:lower:] [:upper:] | tr $KEYFINAL $P`
fi
```

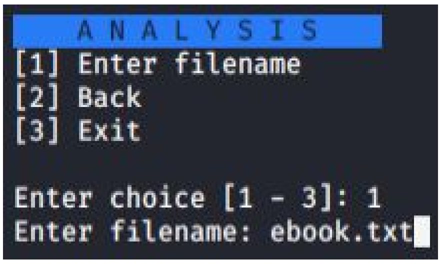
This script does either encryption or decryption according to the passed type.

Analysis and Display

Manual

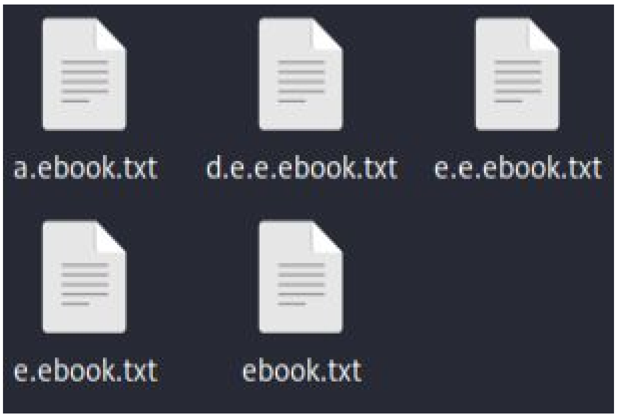
The following manual shows the steps to do analysis and draw a conclusion based on the outputs. The display function is used to observe the outputs.

<table>
<tr><th>Step</th><th>Interface</th><th>Description</th></tr>
<tr><td>1</td><td>

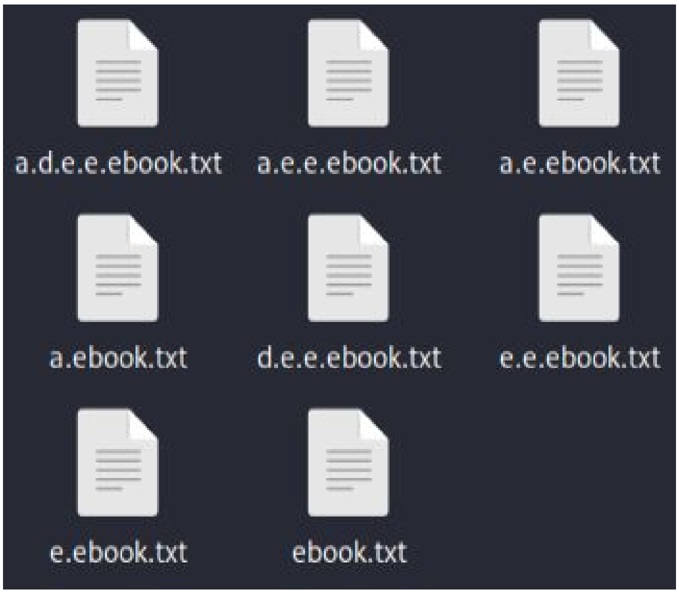


```
    A N A L Y S I S
[1] Enter filename
[2] Back
[3] Exit

Enter choice [1 - 3]: 1
Enter filename: ebook.txt
```

</td><td>After selecting [3] for 'Analysis' in the 'MAIN-MENU', the next page is the 'ANALYSIS' menu. Choice of '1' is entered to enter a filename and 'ebook.txt' as filename.</td></tr>
<tr><td>2</td><td>



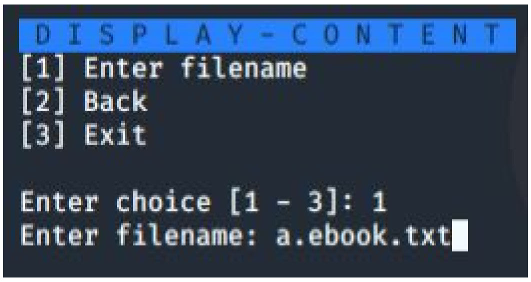
</td><td>Now, there is a new text file 'a.ebook.txt' in the test_files folder. The prefix of 'a.' indicates that it is an analysis of the proceeding filename.</td></tr>
<tr><td>3</td><td>



</td><td>Step 1 is repeated for the filenames:
- e.ebook.txt
- e.e.ebook.txt

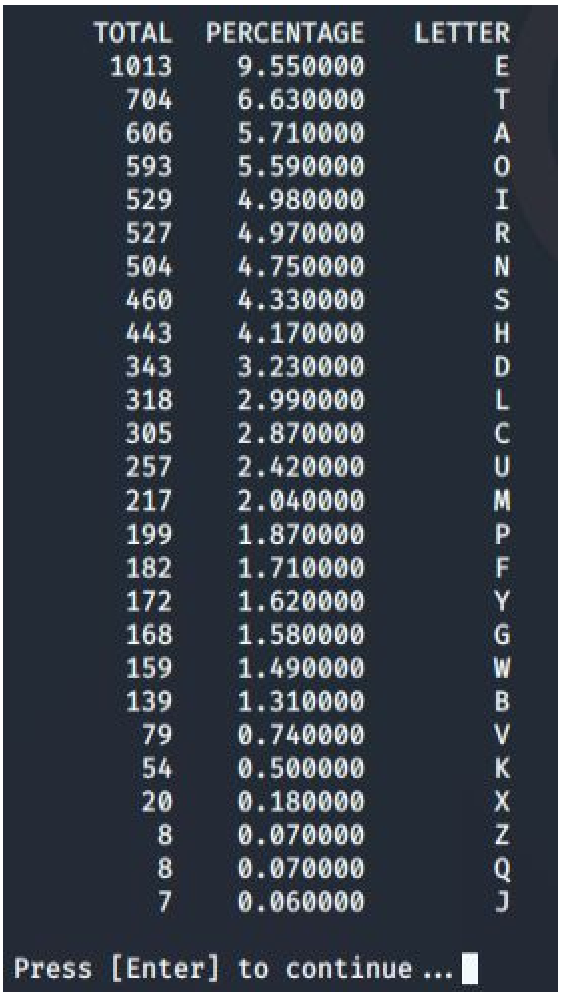
The respective resulting files are:
- a.e.ebook.txt
- a.e.e.ebook.txt</td></tr>
</table>

<table>
<tr>
<td>4</td>
<td>



```
D I S P L A Y - C O N T E N T
[1] Enter filename
[2] Back
[3] Exit

Enter choice [1 - 3]: 1
Enter filename: a.ebook.txt
```

</td>
<td>Navigate back to the ‘MAIN-MENU’ and select [4] to go to the ‘DISPLAY-CONTENT’ menu. Choice of ‘1’ is entered to enter a filename and ‘a.ebook.txt’ as filename.</td>
</tr>
<tr>
<td>5</td>
<td>



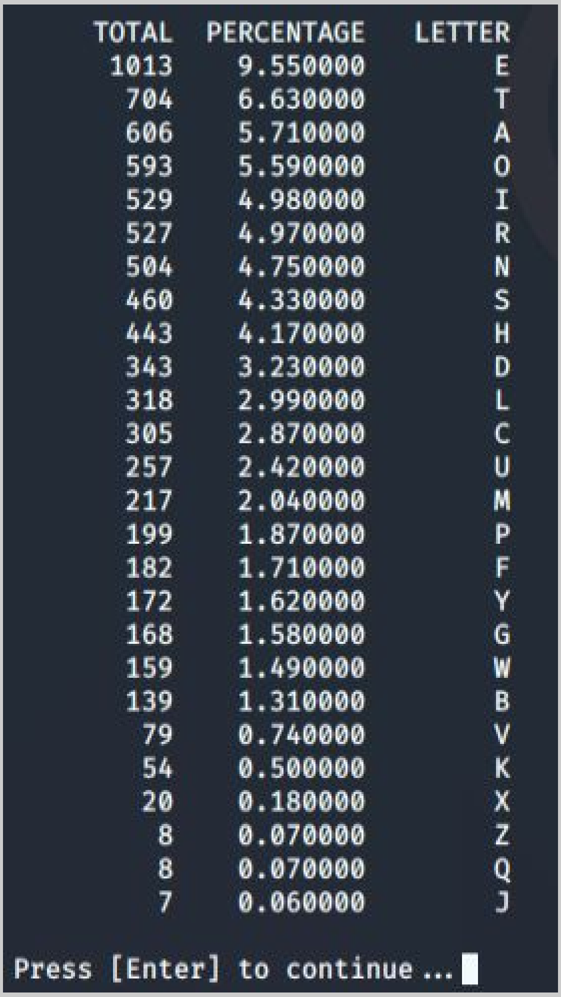
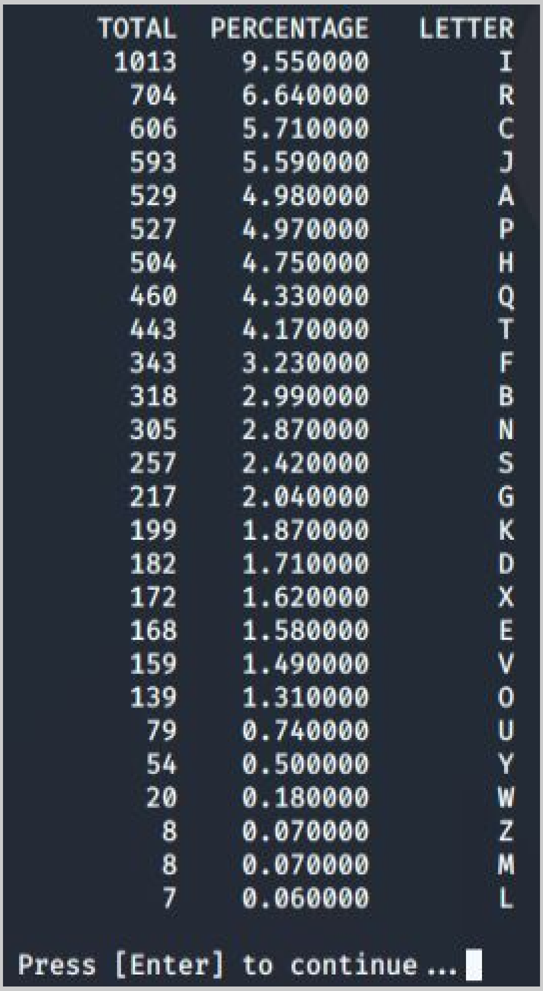
```
      TOTAL   PERCENTAGE    LETTER
       1013     9.550000         E
        704     6.630000         T
        606     5.710000         A
        593     5.590000         O
        529     4.980000         I
        527     4.970000         R
        504     4.750000         N
        460     4.330000         S
        443     4.170000         H
        343     3.230000         D
        318     2.990000         L
        305     2.870000         C
        257     2.420000         U
        217     2.040000         M
        199     1.870000         P
        182     1.710000         F
        172     1.620000         Y
        168     1.580000         G
        159     1.490000         W
        139     1.310000         B
         79     0.740000         V
         54     0.500000         K
         20     0.180000         X
          8     0.070000         Z
          8     0.070000         Q
          7     0.060000         J

Press [Enter] to continue ...
```

</td>
<td>Now, the contents of ‘a.ebook.txt’ are displayed.

Step 4 is repeated with the following filenames:
- a.e.ebook.txt
- a.e.e.ebook.txt</td>
</tr>
</table>

Results and Conclusions

Result 1

a.ebook.txt	a.e.ebook.txt
	

```
     TOTAL   PERCENTAGE   LETTER
      1013     9.550000        E
       704     6.630000        T
       606     5.710000        A
       593     5.590000        O
       529     4.980000        I
       527     4.970000        R
       504     4.750000        N
       460     4.330000        S
       443     4.170000        H
       343     3.230000        D
       318     2.990000        L
       305     2.870000        C
       257     2.420000        U
       217     2.040000        M
       199     1.870000        P
       182     1.710000        F
       172     1.620000        Y
       168     1.580000        G
       159     1.490000        W
       139     1.310000        B
        79     0.740000        V
        54     0.500000        K
        20     0.180000        X
         8     0.070000        Z
         8     0.070000        Q
         7     0.060000        J

Press [Enter] to continue ...
```

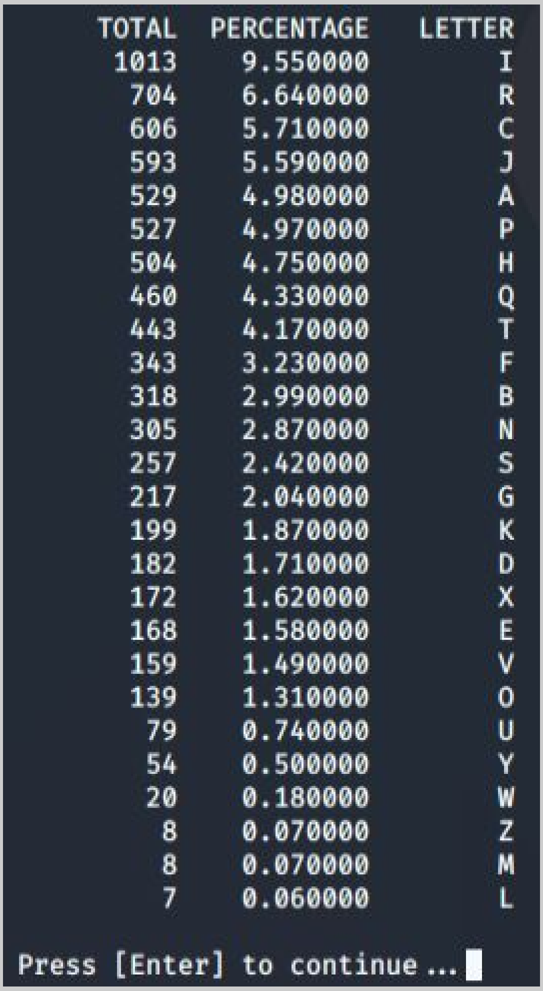
```
     TOTAL   PERCENTAGE   LETTER
      1013     9.550000        I
       704     6.640000        R
       606     5.710000        C
       593     5.590000        J
       529     4.980000        A
       527     4.970000        P
       504     4.750000        H
       460     4.330000        Q
       443     4.170000        T
       343     3.230000        F
       318     2.990000        B
       305     2.870000        N
       257     2.420000        S
       217     2.040000        G
       199     1.870000        K
       182     1.710000        D
       172     1.620000        X
       168     1.580000        E
       159     1.490000        V
       139     1.310000        O
        79     0.740000        U
        54     0.500000        Y
        20     0.180000        W
         8     0.070000        Z
         8     0.070000        M
         7     0.060000        L

Press [Enter] to continue ...
```

Conclusion

It can be concluded that the most occurring letters in the plaintext ‘ebook.txt’ is still reappearing after the first round of encryption in the ciphertext ‘e.ebook.txt’.

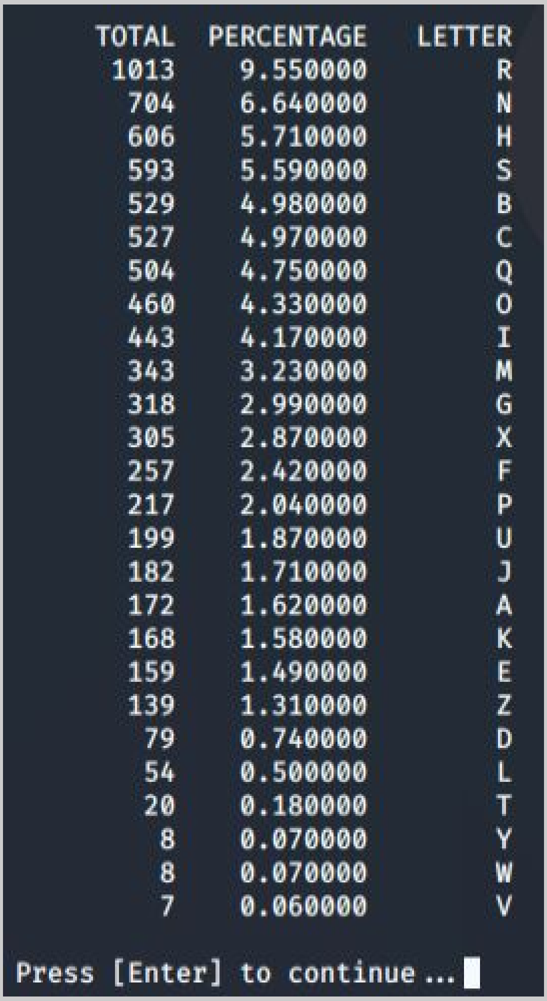
Result 2

a.e.ebook.txt



```
   TOTAL   PERCENTAGE    LETTER
   1013      9.550000         I
    704      6.640000         R
    606      5.710000         C
    593      5.590000         J
    529      4.980000         A
    527      4.970000         P
    504      4.750000         H
    460      4.330000         Q
    443      4.170000         T
    343      3.230000         F
    318      2.990000         B
    305      2.870000         N
    257      2.420000         S
    217      2.040000         G
    199      1.870000         K
    182      1.710000         D
    172      1.620000         X
    168      1.580000         E
    159      1.490000         V
    139      1.310000         O
     79      0.740000         U
     54      0.500000         Y
     20      0.180000         W
      8      0.070000         Z
      8      0.070000         M
      7      0.060000         L

Press [Enter] to continue ...
```

a.e.e.ebook.txt



```
   TOTAL   PERCENTAGE    LETTER
   1013      9.550000         R
    704      6.640000         N
    606      5.710000         H
    593      5.590000         S
    529      4.980000         B
    527      4.970000         C
    504      4.750000         Q
    460      4.330000         O
    443      4.170000         I
    343      3.230000         M
    318      2.990000         G
    305      2.870000         X
    257      2.420000         F
    217      2.040000         P
    199      1.870000         U
    182      1.710000         J
    172      1.620000         A
    168      1.580000         K
    159      1.490000         E
    139      1.310000         Z
     79      0.740000         D
     54      0.500000         L
     20      0.180000         T
      8      0.070000         Y
      8      0.070000         W
      7      0.060000         V

Press [Enter] to continue ...
```

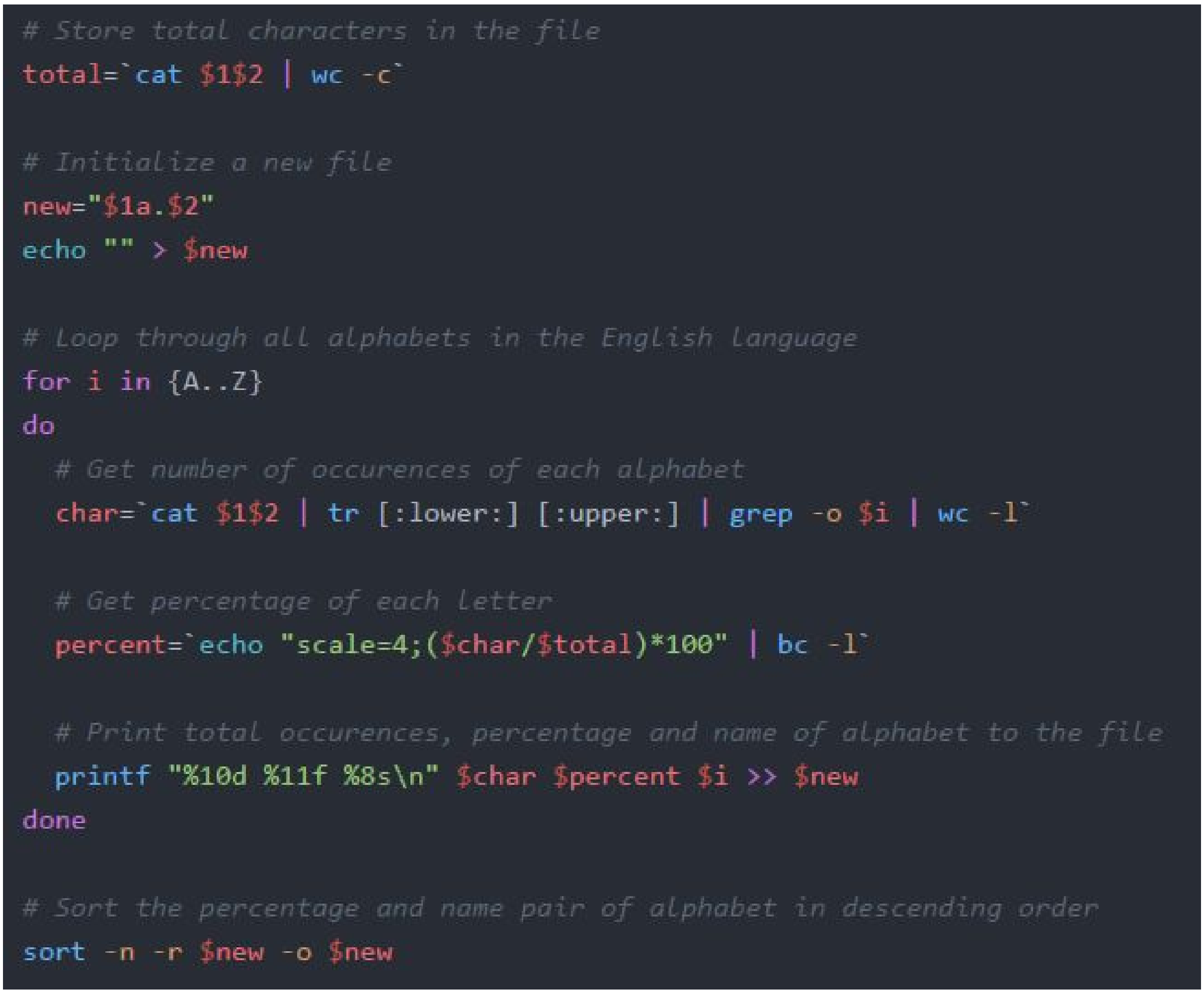
Conclusion

It can be concluded that the strength of multiple rounds of encryption is equivalent to a single round of encryption. This is because the shift cipher creates a single mapping between the alphabets in the source file and the alphabets with the keyword. Having a single mapping retains the frequency of letters. Thus, multiple rounds of encryption does not make the text harder to hack.

Implementation

After selecting [3] for ‘Analysis’ in the ‘MAIN-MENU’, the next page is the ‘ANALYSIS’ menu which is a loop of getting filename to be analyzed. This function is performed by the ‘do_analysis.sh’ script.

do_analysis.sh



```
# Store total characters in the file
total=`cat $1$2 | wc -c`

# Initialize a new file
new="$1a.$2"
echo "" > $new

# Loop through all alphabets in the English language
for i in {A..Z}
do
  # Get number of occurences of each alphabet
  char=`cat $1$2 | tr [:lower:] [:upper:] | grep -o $i | wc -l`

  # Get percentage of each letter
  percent=`echo "scale=4;($char/$total)*100" | bc -l`

  # Print total occurences, percentage and name of alphabet to the file
  printf "%10d %11f %8s\n" $char $percent $i >> $new
done

# Sort the percentage and name pair of alphabet in descending order
sort -n -r $new -o $new
```

After the filename is obtained from the folder test_files, the code snippet above is run with arguments $1 being the directory and $2 as the filename to be analyzed.